Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

**Лабораторная работа №5-7 по курсу**

**«Операционные системы»**

Группа: М8О-206Б-22

Студент: Ларин И.А.

Преподаватель: Миронов Е.С.

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

Дата: 14.12.2023

Москва, 2023

**Постановка задачи**

**Вариант 44.**

Реализовать распределенную систему по асинхронной обработке запросов. В даннойраспределенной системе должно существовать 2 вида узлов: «управляющий» и«вычислительный». Необходимо объединить данные узлы в соответствии с той топологией, которая определена вариантом. Связь между узлами необходимо осуществить при помощитехнологии очередей сообщений. Также в данной системе необходимо предусмотреть проверку доступности узлов в соответствии с вариантом. При убийстве («kill -9») 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.

Список основных поддерживаемых команд:

Создание нового вычислительного узла

Формат команды: create id [parent]

id – целочисленный идентификатор нового вычислительного узла

parent – целочисленный идентификатор родительского узла. Если топологией не предусмотрено введение данного параметра, то его необходимо игнорировать (если его ввели)

Формат вывода:

«Ok: pid», где pid – идентификатор процесса для созданного вычислительного узла

«Error: Already exists» - вычислительный узел с таким идентификатором уже существует

«Error: Parent not found» - нет такого родительского узла с таким идентификатором

«Error: Parent is unavailable» - родительский узел существует, но по каким-то причинам с ним не

удается связаться

«Error: [Custom error]» - любая другая обрабатываемая ошибка

Пример:

> create 10 5

Ok: 3128

Примечания: создание нового управляющего узла осуществляется пользователем программы при помощи запуска исполняемого файла. Id и pid — это разные идентификаторы.

Исполнение команды на вычислительном узле

Формат команды: exec id [params]

id – целочисленный идентификатор вычислительного узла, на который отправляется команда

Формат вывода:

«Ok:id: [result]», где result – результат выполненной команды

«Error:id: Not found» - вычислительный узел с таким идентификатором не найден

«Error:id: Node is unavailable» - по каким-то причинам не удается связаться с вычислительным

узлом

«Error:id: [Custom error]» - любая другая обрабатываемая ошибка

Пример:

Можно найти в описании конкретной команды, определенной вариантом задания.

Примечание: выполнение команд должно быть асинхронным. Т.е. пока выполняется команда наодном из вычислительных узлов, то можно отправить следующую команду на другой вычислительный узел.

Топология 4

Аналогично топологии 4, но узлы находятся в идеально сбалансированном бинарном дереве. Каждый следующий узел должен добавляться в самое наименьшее поддерево.

Типы команд для вычислительных узлов

Набор команд 1 (подсчет суммы n чисел)

Формат команды: exec id n k1 … kn

id – целочисленный идентификатор вычислительного узла, на который отправляется команда

n – количество складываемых чисел (от 1 до 108)

k1 … kn – складываемые числа

Пример:

> exec 10 3 1 2 3

Ok:10: 6

Тип проверки доступности узлов

Команда проверки 3

Формат команды: heartbit time

Каждый узел начинает сообщать раз в time миллисекунд о том, что он работоспособен. Если от

узла нет сигнала в течении 4\*time миллисекунд, то должна выводится пользователю строка:

«Heartbit: node id is unavailable now», где id – идентификатор недоступного вычислительного узла.

Пример:

> heartbit 2000

Ok

Пример:

> ping 10

Ok: 1 // узел 10 доступен

> ping 17

Ok: 0 // узел 17 недоступен

**Общий метод и алгоритм решения**

Программа состоит из нескольких компонентов, включая вычислительные, управляющие узлы и структуру клиент-серверной архитектуры. При создании нового вычислительного узла, который будет добавлен в дерево, будет запущен новый процесс. Взаимодействие между узлами будет осуществляться с использованием брокера сообщений, в данном случае - zero message queue. Будет разработан интерфейс командной строки для интерактивного общения с пользователем. При удалении узла будут также удалены все его потомки. В заданные интервалы времени каждый узел должен регулярно отправлять сообщение о своей работоспособности с использованием команды heartbit.

**Код программы**

Client.cpp

#include <iostream>

#include "topology.h"

#include <set>

#include <limits>

#include "myZmq.h"

#include <sys/time.h>

#include <string>

#include <unistd.h>

#include "tree.h"

std::vector<int> treee;

int main(){

    std::string command;

    TNode node(-1, -1, -1);// cоздаём узел (может управляемый) у него id = -1, parentPort = -1, parentID = -1, leftID and right = -2 (пока не существует)

    Tree tree;

    std::string answer;

    while(std::cin >> command){

        if(command == "create"){

            int child;

            std::cin >> child; // индификатор дочернего

            if(tree.exist(child)){

                std::cout << "Error: child already existed!\n";

            }

            else{

                treee.push\_back(child); // запоминаем в вектор

                while(true){

                    int idParent = tree.findId(); // находим родительское id

                    if(idParent == node.id){ // создаём в начале, так как первое включение

                        answer = node.createChild(child);// id ребёнка

                        tree.addElem(child, idParent);

                        break;

                    }

                    else{

                        std::string message = "create " + std::to\_string(child);

                        answer = node.sendStr(message, idParent);// определили, для какого родителя свободно место

                        if(answer == "Error: id is not found"){

                            tree.notAvailable(idParent);// можно удалить скорее всего

                        }

                        else{

                            tree.addElem(child, idParent);

                            break;

                        }

                    }

                }

                std::cout << answer << std::endl;

            }

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

            int time;

            std::cin >> time;

            for (int i = 0; i <= 2; ++i) {

                for (int childC = 0; childC < treee.size(); ++childC) {

                    if(!tree.exist(treee[childC])){

                        std::cout << "Error: child is not existed!\n";

                    }

                    else if(node.leftId == treee[childC] || node.rightId == treee[childC]){

                        answer = node.Ping(treee[childC]);

                        std::cout << "(" << treee[childC] << ") => " << answer << std::endl;

                    }

                    else{

                        std::string message = "ping " + std::to\_string(treee[childC]);

                        answer = node.sendStr(message, treee[childC]);

                        if(answer == "Error: id is not found"){

                            answer = "OK: 0";

                        }

                        std::cout << "(" << treee[childC] << ") => " << answer << std::endl;

                    }

                }

                usleep(time);

            }

        } else if(command == "ping"){

            int child;

            std::cin >> child;

            if(!tree.exist(child)){

                std::cout << "Error: child is not existed!\n";

            }

            else if(node.leftId == child || node.rightId == child){

                answer = node.Ping(child);

                std::cout << answer << std::endl;

            }

            else{

                std::string message = "ping " + std::to\_string(child);

                answer = node.sendStr(message, child);

                if(answer == "Error: id is not found"){

                    answer = "OK: 0";

                }

                std::cout << answer << std::endl;

            }

        } else if(command == "exec"){

            std::string str;

            int child;

            std::cin >> child;

            getline(std::cin, str);

            if(!tree.exist(child)){

                std::cout << "Error: child is not existed!\n";

            }else{

                std::string message = "exec " + str;

                answer = node.sendStr(message, child);

                std::cout << answer << std::endl;

            }

        }

        else if(command == "remove"){

            int child;

            std::cin >> child;

            std::string message = "remove";

            if(!tree.exist(child)){

                std::cout << "Error: child is not existed!\n";

            }else{

                answer = node.sendStr(message, child);//удали такого 1

                treee.erase(std::remove(treee.begin(), treee.end(), child), treee.end());// сначала удаляются все дальние потомки кроме самых ближних если они рядом с началом то удаляются здесть, если далеко то будет функция

                if(answer != "Error: id is not found"){

                    tree.Remove(child);

                    if(child == node.leftId){

                        unbind(node.left, node.leftPort);

                        node.leftId = -2;

                        answer = "OK";

                    }

                    else if(child == node.rightId){

                        node.rightId = -2;

                        unbind(node.right, node.rightPort);

                        answer = "OK";

                    }

                    else{

                        message = "clear " + std::to\_string(child);

                        answer = node.sendStr(message, std::stoi(answer));

                    }

                    std::cout << answer << std::endl;

                }

            }

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

            node.removeElem();

            tree.Remove(tree.findId());

            treee.clear();

            break;

        }

    }

    return 0;

}

Server.cpp

#include <bits/stdc++.h>

#include "topology.h"

#include <sys/time.h>

#include <set>

#include "myZmq.h"

#include <string>

#include "tree.h"

#include <sstream>

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

    if (argc < 4){

        printf("Usage: ./main childID parentPort parentID\n");

        exit(-1);

    }

    TNode node(atoi(argv[1]), atoi(argv[2]), atoi(argv[3]));

    while(true){

        std::string message;

        std::string command;

        message = receive\_message(node.parent);

        std::istringstream request(message);

        request >> command;

        if(command == "pid"){

            std::string answer = std::to\_string(getpid());

            send\_message(node.parent, answer);

        }

        else if(command == "ping"){

            int child;

            request >> child;

            std::string answer = node.Ping(child);

            send\_message(node.parent, answer);

        }

        else if(command == "create"){

            int child;

            request >> child;

            std::string answer = node.createChild(child);

            send\_message(node.parent, answer);

        }

        else if(command == "send"){

            int child;

            std::string str;

            request >> child;

            getline(request, str);

            str.erase(0, 1);// Зачем?

            std::string answer = node.sendStr(str, child);

            send\_message(node.parent, answer);

        }

        else if(command == "exec"){

            std::string str;

            getline(request, str);

            std::string answer = node.exec(str);

            send\_message(node.parent, answer);

        }

        else if(command == "remove"){

            std::string answer = node.removeElem();// перешли к левому ближнему 3

            send\_message(node.parent, answer);// отправлеам родительский и отсоединяемся от родителя

            disconnect(node.parent, node.parentPort);

            node.parent.close();

            break;

        }else if(command == "clear"){

            int child;

            request >> child;

            std::string answer = node.clearChild(child);

            send\_message(node.parent, answer);

        }

    }

    sleep(1);

    return 0;

}

myZmq.h

#pragma once

#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) {

        throw -1;

    }

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

    return received\_msg;

}

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

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

    socket.connect(adress);

}

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

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

    socket.disconnect(adress);

}

// привязывает сокет к определённому порту

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

    int port = MAIN\_PORT + id;

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

    while(1){

        try{

            socket.bind(adress);

            break;

        }

        catch(...){

            port++;

        }

    }

    return port;

}

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

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

    socket.unbind(adress);

}

Static\_main.c

#include "realization.h"

#include <stdio.h>

int main() {

  int c = 0;

  while(scanf("%d", &c) != EOF) {

      if(c == 1){

        float point, increment;

        if(scanf("%f %f", &point, &increment) == 2) {

          printf("%.6f\n", derivative(point, increment));

        }

      }

      else if(c ==2){

        long numeric;

        if(scanf("%ld", &numeric) == 1) {

          printf("%s\n", translation(numeric));

        }

      }

      else{

        printf("Invalid\n");

      }

    }

  return 0;

}

Topology.h

#include <iostream>

#include "myZmq.h"

#include <string>

#include <sstream>

#include "unistd.h"

class TNode{

    private:

        zmq::context\_t context;

    public:

        zmq::socket\_t left, right, parent;// в инициализации задаём одни контексты, чтобы взаимодействовали между собой.

        int leftPort, rightPort, parentPort;

        int id, leftId = -2, rightId = -2, parentId;

        TNode(int idElem, int parentP, int parID): id(idElem), parentPort(parentP), parentId(parID),

        left(context,ZMQ\_REQ), right(context,ZMQ\_REQ), parent(context,ZMQ\_REP){

            if(idElem != -1){

                connect(parent, parentPort);// если есть родитель, то его нужно связать с родительским портом (-1 в самом начале получается при инициализации)

            }

        }

        ~TNode(){}

        std::string createChild(int childId){

            int port;

            bool isleft = false;

            if(leftId == -2){

                leftPort = bind(left, childId); //номер порта, к которому привязали left id

                leftId = childId; //запоминаем left id

                port = leftPort; // засовываем в переменну left port

                isleft = true; // занят

            }else if(rightId == -2){

                rightPort = bind(right, childId);

                rightId = childId;

                port = rightPort;

            }else{

                return "Error create";

            }

            int pid = fork();

            if(pid == 0){

                execl("./server", "server", std::to\_string(childId).c\_str(), std::to\_string(port).c\_str(), std::to\_string(id).c\_str(), (char\*)NULL);// id = -1 - родительский

                std::cout << "Error execl" << std::endl;

                exit(1);

            }

            else{

                std::string pidChild;

                try{

                    if(isleft){

                        left.setsockopt(ZMQ\_SNDTIMEO, 3000);

                        send\_message(left, "pid");

                        pidChild = receive\_message(left);

                    }

                    else{

                        right.setsockopt(ZMQ\_SNDTIMEO, 3000);

                        send\_message(right, "pid");

                        pidChild = receive\_message(right);

                    }

                    return "OK: " + pidChild;

                }

                catch(int){

                    return "Error: can't connect to child";

                }

            }

        }

        std::string Ping(int idElem){

            std::string answer = "OK: 0";

            if(idElem == id){

                answer = "OK: 1";

                return answer;

            }

            else if(leftId == idElem){

                std::string message = "ping " + std::to\_string(idElem);

                send\_message(left, message);

                try{

                    message = receive\_message(left);

                    if(message == "OK: 1"){

                        answer = message;

                    }

                }

                catch(int){}

            }

            else if(rightId == idElem){

                std::string message = "ping " + std::to\_string(idElem);

                send\_message(right, message);

                try{

                    message = receive\_message(right);

                    if(message == "OK: 1"){

                        answer = message;

                    }

                }

                catch(int){}

            }

            return answer;

        }

        std::string sendStr(std::string str, int idElem){

            std::string answer = "Error: id is not found";

            if(leftId == -2 && rightId == -2){

                return answer;

            }

            else if(leftId == idElem){

                if(Ping(leftId) == "OK: 1"){

                    send\_message(left, str);//оказался левым ближним 2

                    try{

                        answer = receive\_message(left);

                    }

                    catch(int){}

                }

            }

            else if(rightId == idElem){

                if(Ping(rightId) == "OK: 1"){

                    send\_message(right, str);

                    try{

                        answer = receive\_message(right);

                    }

                    catch(int){}

                }

            }

            else{

                if(Ping(leftId) == "OK: 1"){

                    std::string message = "send " + std::to\_string(idElem) + " " + str;

                    send\_message(left, message);

                    try{

                        message = receive\_message(left);

                    }

                    catch(int){

                        message = "Error: id is not found";

                    }

                    if(message != "Error: id is not found"){

                        answer = message;

                    }

                }

                if(Ping(rightId) == "OK: 1"){

                    std::string message = "send " + std::to\_string(idElem) + " " + str;

                    send\_message(right, message);

                    try{

                        message = receive\_message(right);

                    }

                    catch(int){

                        message = "Error: id is not found";

                    }

                    if(message != "Error: id is not found"){

                        answer = message;

                    }

                }

            }

            return answer;

        }

        std::string exec(std::string str){

            std::istringstream numbers(str);

            int sum = 0;

            int count, number;

            numbers >> count;

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

                numbers >> number;

                sum += number;

            }

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

            return answer;

        }

        std::string clearChild(int child){

            if(leftId == child){

                leftId = -2;

                unbind(left, leftPort);

            }

            else{

                rightId = -2;

                unbind(right, rightPort);

            }

            return "OK";

        }

        std::string removeElem(){

            //std::string answer;

            if(leftId != -2){

                if(Ping(leftId) == "OK: 1"){// удаляем потомков левого ближнего 4

                    std::string message = "remove";

                    send\_message(left, message);

                    try{

                        message = receive\_message(left);

                    }

                    catch(int){}

                    unbind(left, leftPort);

                    left.close();

                }

            }

            if(rightId != -2){

                if(Ping(rightId) == "OK: 1"){

                    std::string message = "remove";

                    send\_message(right, message);

                    try{

                        message = receive\_message(right);

                    }

                    catch(int){}

                    unbind(right, rightPort);

                    right.close();

                }

            }

            return std::to\_string(parentId);

        }

};

Tree.h

#include <bits/stdc++.h>

#include<vector>

#include <set>

#include <algorithm>

class Tree{

    struct Node{

            int id;

            bool available;

            int height;

            Node\* left, \*right;

            Node(int i): id(i), available(true), left(nullptr), right(nullptr){}

            ~Node(){}

            void findRemove(int idElem, std::set<int> &ids){

                if(left != nullptr && left->id == idElem){// нашли нужный нам элемент, удаляем потомков с нужным элеметом

                    left->removeChilds(ids);

                    ids.erase(left->id);

                    delete left;

                    left = nullptr;

                }

                else if(right != nullptr && right->id == idElem){

                    right->removeChilds(ids);

                    ids.erase(right->id);

                    delete right;

                    right = nullptr;

                }

                else{

                    if(left != nullptr){// ищем нужный нам элнмент

                        left->findRemove(idElem, ids);

                    }

                    if(right != nullptr){

                        right->findRemove(idElem, ids);

                    }

                }

            }

            void removeChilds(std::set<int> &ids){

                if(left != nullptr){

                    left->removeChilds(ids);

                    ids.erase(left->id);

                    delete left;

                    left = nullptr;

                }

                if(right != nullptr){

                    right->removeChilds(ids);

                    ids.erase(right->id);

                    delete right;

                    right = nullptr;

                }

            }

            void addElem(int idElem, int parent, std::set<int> &ids){

                if(id == parent){//если нашли нужный

                    if(left == nullptr){

                        left = new Node(idElem);

                    }

                    else{

                        right = new Node(idElem);

                    }

                    ids.insert(idElem);

                }

                else{

                    if(left != nullptr){//не нашили идём по следующим узлам

                        left->addElem(idElem, parent, ids);

                    }

                    if(right != nullptr){

                        right->addElem(idElem, parent, ids);

                    }

                }

            }

            int minHight(){

                if(left == nullptr || right == nullptr){

                    return 0;

                }

                int lHight = -1;

                int rHight = -1;

                if(left != nullptr && left->available == true){

                    lHight = left->minHight();

                }

                if(right != nullptr && right->available == true){

                    rHight = right->minHight();

                }

                if(rHight == -1 && lHight == -1){

                    available = false;

                    return -1;

                }

                else if(rHight  == -1){

                    return lHight + 1;

                }

                else if(lHight == -1){

                    return rHight + 1;

                }

                else{

                    return std::min(lHight, rHight) + 1;

                }

            }

            void notAvailable(int idElem){

                if(id == idElem){

                    available = false;

                }

                else{

                    if(left != nullptr){

                        left->notAvailable(idElem);

                    }

                    if(right != nullptr){

                        right->notAvailable(idElem);

                    }

                }

            }

            int idForMinH(int hight, int curHight){

                if(hight < curHight){

                    return -2;

                }

                else if(hight > curHight){

                    int curid = -2;

                    if(left != nullptr && left->available == true){

                        curid = left->idForMinH(hight, (curHight + 1));

                    }

                    if(right != nullptr && right->available == true && curid == -2){

                        curid = right->idForMinH(hight, (curHight + 1));

                    }

                    return curid;

                }

                else{

                    if(left == nullptr || right == nullptr){

                        return id;

                    }

                    return -2;

                }

            }

    };

    private:

        Node\* root;

    public:

        std::set<int> ids;

        Tree(){

            root = new Node(-1);

        }

        bool exist(int id){

            if(ids.find(id) != ids.end()){

                return true;

            }

            return false;

        }

        void notAvailable(int id){

            root->notAvailable(id);

        }

        int findId(){

            int h = root->minHight();

            return root->idForMinH(h, 0);

        }

        void addElem(int id, int parent){

            root->addElem(id, parent, ids);

        }

         void Remove(int idElem){

            root->findRemove(idElem, ids);

         }

        ~Tree(){

            root->removeChilds(ids);

            delete root;

        }

};

**Протокол работы программы**

**Тестирование:**

ilya@ilya-ilyal:~/osi5-7$ ./client

create 1

OK: 4085

create 5

OK: 4106

create 3

OK: 4151

ping 1

OK: 1

remove 3

OK

ping 3

Error: child is not existed!

heartbit 2000

(1) => OK: 1

(5) => OK: 1

(1) => OK: 1

(5) => OK: 1

(1) => OK: 1

(5) => OK: 1

exec 5 3 1 1 1

OK: 5: 3

kill

**Strace:**

**Часть Strace, полный в file.txt**

execve("./client", ["./client"], 0x7ffcda879a68 /\* 59 vars \*/) = 0

brk(NULL) = 0x55f7906f6000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fff7f67f9b0) = -1 EINVAL (Недопустимый аргумент)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f112d15a000

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

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=54935, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 54935, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f112d14c000

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\240\233\1\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=634936, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 636784, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112d0b0000

mmap(0x7f112d0c8000, 397312, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7f112d0c8000

mmap(0x7f112d129000, 106496, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x79000) = 0x7f112d129000

mmap(0x7f112d143000, 36864, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x92000) = 0x7f112d143000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=2260296, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 2275520, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112ce00000

mprotect(0x7f112ce9a000, 1576960, PROT\_NONE) = 0

mmap(0x7f112ce9a000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7f112ce9a000

mmap(0x7f112cfab000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7f112cfab000

mmap(0x7f112d01b000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7f112d01b000

mmap(0x7f112d029000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f112d029000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=125488, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 127720, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112d090000

mmap(0x7f112d093000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f112d093000

mmap(0x7f112d0aa000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7f112d0aa000

mmap(0x7f112d0ae000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7f112d0ae000

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\0P\237\2\0\0\0\0\0"..., 832) = 832

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

pread64(3, "\4\0\0\0 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"..., 48, 848) = 48

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\244;\374\204(\337f#\315I\214\234\f\256\271\32"..., 68, 896) = 68

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2216304, ...}, AT\_EMPTY\_PATH) = 0

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

mmap(NULL, 2260560, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112ca00000

mmap(0x7f112ca28000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7f112ca28000

mmap(0x7f112cbbd000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7f112cbbd000

mmap(0x7f112cc15000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x214000) = 0x7f112cc15000

mmap(0x7f112cc1b000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f112cc1b000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libbsd.so.0", O\_RDONLY|O\_CLOEXEC) = 3

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=89096, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 94432, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112d078000

mprotect(0x7f112d07c000, 69632, PROT\_NONE) = 0

mmap(0x7f112d07c000, 53248, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f112d07c000

mmap(0x7f112d089000, 12288, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x11000) = 0x7f112d089000

mmap(0x7f112d08d000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x14000) = 0x7f112d08d000

mmap(0x7f112d08f000, 224, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f112d08f000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=355040, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 357440, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112cda8000

mprotect(0x7f112cdb4000, 303104, PROT\_NONE) = 0

mmap(0x7f112cdb4000, 229376, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xc000) = 0x7f112cdb4000

mmap(0x7f112cdec000, 69632, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x44000) = 0x7f112cdec000

mmap(0x7f112cdfe000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x55000) = 0x7f112cdfe000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f112d076000

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libpgm-5.3.so.0", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\340L\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=310264, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 329808, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112cd57000

mmap(0x7f112cd5b000, 172032, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f112cd5b000

mmap(0x7f112cd85000, 118784, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2e000) = 0x7f112cd85000

mmap(0x7f112cda2000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4a000) = 0x7f112cda2000

mmap(0x7f112cda4000, 14416, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f112cda4000

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 \255\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=497824, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 1223168, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112cc2c000

mprotect(0x7f112cc36000, 446464, PROT\_NONE) = 0

mmap(0x7f112cc36000, 286720, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7f112cc36000

mmap(0x7f112cc7c000, 155648, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x50000) = 0x7f112cc7c000

mmap(0x7f112cca3000, 16384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x76000) = 0x7f112cca3000

mmap(0x7f112cca7000, 719360, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f112cca7000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=338648, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 340960, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112c9ac000

mprotect(0x7f112c9b7000, 282624, PROT\_NONE) = 0

mmap(0x7f112c9b7000, 229376, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb000) = 0x7f112c9b7000

mmap(0x7f112c9ef000, 49152, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x43000) = 0x7f112c9ef000

mmap(0x7f112c9fc000, 16384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4f000) = 0x7f112c9fc000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=940560, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 942344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112c8c5000

mmap(0x7f112c8d3000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7f112c8d3000

mmap(0x7f112c94f000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7f112c94f000

mmap(0x7f112c9aa000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7f112c9aa000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libmd.so.0", O\_RDONLY|O\_CLOEXEC) = 3

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=47472, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 49384, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112d069000

mmap(0x7f112d06b000, 28672, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f112d06b000

mmap(0x7f112d072000, 8192, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9000) = 0x7f112d072000

mmap(0x7f112d074000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7f112d074000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=21448, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f112d067000

mmap(NULL, 16424, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112d062000

mmap(0x7f112d063000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7f112d063000

mmap(0x7f112d064000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f112d064000

mmap(0x7f112d065000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f112d065000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=827936, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 830576, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112c7fa000

mprotect(0x7f112c81b000, 634880, PROT\_NONE) = 0

mmap(0x7f112c81b000, 380928, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21000) = 0x7f112c81b000

mmap(0x7f112c878000, 249856, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x7e000) = 0x7f112c878000

mmap(0x7f112c8b6000, 61440, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xbb000) = 0x7f112c8b6000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=182864, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 188472, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112d033000

mprotect(0x7f112d037000, 163840, PROT\_NONE) = 0

mmap(0x7f112d037000, 110592, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f112d037000

mmap(0x7f112d052000, 49152, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1f000) = 0x7f112d052000

mmap(0x7f112d05f000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2b000) = 0x7f112d05f000

mmap(0x7f112d061000, 56, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f112d061000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=18504, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 20552, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112d02d000

mmap(0x7f112d02f000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f112d02f000

mmap(0x7f112d030000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f112d030000

mmap(0x7f112d031000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f112d031000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=52016, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 54224, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112c7ec000

mprotect(0x7f112c7ef000, 36864, PROT\_NONE) = 0

mmap(0x7f112c7ef000, 24576, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f112c7ef000

mmap(0x7f112c7f5000, 8192, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9000) = 0x7f112c7f5000

mmap(0x7f112c7f8000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb000) = 0x7f112c7f8000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=22600, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f112cc2a000

mmap(NULL, 24592, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112c7e5000

mmap(0x7f112c7e7000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f112c7e7000

mmap(0x7f112c7e9000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f112c7e9000

mmap(0x7f112c7ea000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f112c7ea000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=68552, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 80456, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f112c7d1000

mmap(0x7f112c7d4000, 40960, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f112c7d4000

mmap(0x7f112c7de000, 12288, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xd000) = 0x7f112c7de000

mmap(0x7f112c7e1000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xf000) = 0x7f112c7e1000

mmap(0x7f112c7e3000, 6728, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f112c7e3000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f112cc28000

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f112c7ce000

arch\_prctl(ARCH\_SET\_FS, 0x7f112c7ce9c0) = 0

set\_tid\_address(0x7f112c7cec90) = 7092

set\_robust\_list(0x7f112c7ceca0, 24) = 0

rseq(0x7f112c7cf360, 0x20, 0, 0x53053053) = 0

mprotect(0x7f112cc15000, 16384, PROT\_READ) = 0

mprotect(0x7f112c7e1000, 4096, PROT\_READ) = 0

mprotect(0x7f112c7ea000, 4096, PROT\_READ) = 0

mprotect(0x7f112c7f8000, 4096, PROT\_READ) = 0

mprotect(0x7f112d031000, 4096, PROT\_READ) = 0

mprotect(0x7f112d05f000, 4096, PROT\_READ) = 0

mprotect(0x7f112c8b6000, 53248, PROT\_READ) = 0

mprotect(0x7f112d065000, 4096, PROT\_READ) = 0

mprotect(0x7f112d074000, 4096, PROT\_READ) = 0

mprotect(0x7f112c9aa000, 4096, PROT\_READ) = 0

mprotect(0x7f112c9fc000, 8192, PROT\_READ) = 0

mprotect(0x7f112d0ae000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f112c7cc000

mprotect(0x7f112d01b000, 45056, PROT\_READ) = 0

mprotect(0x7f112cca3000, 12288, PROT\_READ) = 0

mprotect(0x7f112cda2000, 4096, PROT\_READ) = 0

mprotect(0x7f112cdfe000, 4096, PROT\_READ) = 0

mprotect(0x7f112d08d000, 4096, PROT\_READ) = 0

mprotect(0x7f112d143000, 32768, PROT\_READ) = 0

mprotect(0x55f7902c8000, 4096, PROT\_READ) = 0

mprotect(0x7f112d194000, 8192, PROT\_READ) = 0

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

munmap(0x7f112d14c000, 54935) = 0

getrandom("\xef\xae\x09\x8a\xa8\x13\x82\x96", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x55f7906f6000

brk(0x55f790717000) = 0x55f790717000

futex(0x7f112d02977c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

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

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

close(3) = 0

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

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

getdents64(3, 0x55f790707ee0 /\* 21 entries \*/, 32768) = 616

getdents64(3, 0x55f790707ee0 /\* 0 entries \*/, 32768) = 0

close(3) = 0

getpid() = 7092

sched\_getaffinity(7092, 128, [0, 1, 2, 3, 4]) = 8

newfstatat(AT\_FDCWD, "/etc/nsswitch.conf", {st\_mode=S\_IFREG|0644, st\_size=542, ...}, 0) = 0

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

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=542, ...}, AT\_EMPTY\_PATH) = 0

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

read(3, "", 4096) = 0

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=542, ...}, AT\_EMPTY\_PATH) = 0

close(3) = 0

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=54935, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 54935, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f112d14c000

close(3) = 0

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

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls/x86\_64/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/x86\_64/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu", {st\_mode=S\_IFDIR|0755, st\_size=69632, ...}, 0) = 0

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls/x86\_64/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/x86\_64/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu", {st\_mode=S\_IFDIR|0755, st\_size=69632, ...}, 0) = 0

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

newfstatat(AT\_FDCWD, "/lib/tls/x86\_64/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/tls/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/tls/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/tls", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

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

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

newfstatat(AT\_FDCWD, "/usr/lib/tls/x86\_64/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/tls/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/tls/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/tls", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64", 0x7fff7f67c840, 0) = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(AT\_FDCWD, "/usr/lib/

**Вывод**

В результате выполнения этого практического задания я освоил использование взаимодействия клиент-серверного типа с помощью очереди сообщений. Я узнал, как оперировать с таким посредником, как zero message queue, а также получил опыт в построении подобной структуры "сервера" в виде дерева.