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: М80-206Б-22

Студент: Голубев Т.Д.

Преподаватель: Миронов Е.С.

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

Дата: 28.12.2023

Москва, 2023

**Постановка задачи**

**Вариант 6.**

Реализовать распределенную систему по асинхронной обработке запросов.

1. Топология 3 -- Все вычислительные узлы хранятся в бинарном дереве поиска. [parent] — является необязательным параметром.
2. Набор команд 4 (поиск подстроки в строке)
3. Команда проверки 1 (pingall)

**Общий метод и алгоритм решения**

Использованные системные вызовы:

* int socket(int domain, int type, int protocol); -- создает конечную точку соединения
* int connect(int sockfd, const struct sockaddr \*serv\_addr, socklen\_t addrlen); -- инициирует соединение на сокете
* int bind(int sockfd, struct sockaddr \*my\_addr, socklen\_t addrlen); -- привязать имя к сокету
* ssize\_t send(int s, const void \*msg, size\_t len, int flags); -- отправляет сообщения в сокет
* int recv(int s, void \*buf, size\_t len, int flags); -- получить сообщение из сокета
* int setsockopt(int s, int level, int optname, const void \*optval, socklen\_t optlen); -- получить или установить флаги на сокете

Для реализации распределённой системы использовалась библиотека ZeroMQ.

**Код программы**

binary\_tree.h

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147 | #pragma once  #include <compare>  #include <stdexcept>  #include <vector>  **namespace** mysys {  **template** <**class** **T**>  requires std::three\_way\_comparable<T>  **class** **BinaryTree** {  **public:**  BinaryTree() : \_root{nullptr} {}  BinaryTree(**const** BinaryTree& other) {  \_root = \_copy\_tree(other->\_root);  }  BinaryTree(BinaryTree&& other) noexcept {  \_root = other.\_root;  }  ~BinaryTree() noexcept {  \_delete\_tree(\_root);  }  BinaryTree& **operator**=(**const** BinaryTree& rhs) {  \_root = \_copy\_tree(rhs.\_root);  **return** \***this**;  }  BinaryTree& **operator**=(BinaryTree&& rhs) noexcept {  \_root = rhs.\_root;  **return** \***this**;  }  **bool** search(T key) **const** {  \_Node\* found = \_search(\_root, key);  **if** (found == nullptr) **return** false;    **return** true;  }  **void** insert(T key) {  **if** (search(key)) **throw** std::logic\_error("Already exists");  \_root = \_insert(\_root, key);  }  std::vector<T> get\_tops() {  std::vector<T> tops;  \_get\_tops(\_root, tops);  **return** tops;  }  std::vector<T> get\_children(T key) {  \_Node\* found = \_search(\_root, key);  **if** (found == nullptr) **throw** std::logic\_error("Key not found");  std::vector<T> tops;  \_get\_tops(found, tops);  **return** tops;  }  T get\_parent(T key) {  \_Node\* found = \_search\_parent(\_root, nullptr, key);  **if** (found == nullptr) **return** **0**;  **return** found->key;  }  **private:**  **struct** \_Node {  T key;  \_Node\* left;  \_Node\* right;  };  \_Node\* \_root;  \_Node\* **\_new\_node**(T item) {  \_Node\* temp = **new** \_Node;  temp->key = item;  temp->left = temp->right = nullptr;  **return** temp;  }    \_Node\* **\_insert**(\_Node\* node, T key) {  **if** (node == nullptr)  **return** \_new\_node(key);  **if** (key < node->key)  node->left = \_insert(node->left, key);  **else** **if** (key > node->key)  node->right = \_insert(node->right, key);    **return** node;  }  **void** \_delete\_tree(\_Node\* node) noexcept {  **if** (node == nullptr) **return**;  \_Node\* left = node->left;  \_Node\* right = node->right;  **delete** node;  \_delete\_tree(left);  \_delete\_tree(right);  }  \_Node\* \_create\_tree(\_Node\* left, \_Node\* right, T key) {  \_Node\* root = \_new\_node(key);  root->left = left;  root->right = right;  **return** root;  }  \_Node\* \_copy\_tree(\_Node\* other) {  **if** (other == nullptr) **return** nullptr;  \_Node\* root = \_new\_node(other->key);  root->left = \_copy\_tree(other->left);  root->right = \_copy\_tree(other->right);  **return** root;  }  \_Node\* \_search(\_Node\* root, **int** key) **const** {  **if** (root == nullptr || root->key == key)  **return** root;  **if** (root->key < key)  **return** \_search(root->right, key);  **return** **\_search**(root->left, key);  }  \_Node\* \_search\_parent(\_Node\* root, \_Node\* prev, **int** key) **const** {  **if** (root == nullptr || root->key == key)  **return** prev;  **if** (root->key < key)  **return** \_search\_parent(root->right, root, key);  **return** **\_search\_parent**(root->left, root, key);  }  **void** \_get\_tops(\_Node\* root, std::vector<T>& tops) {  **if** (root == nullptr) **return**;  \_get\_tops(root->left, tops);  \_get\_tops(root->right, tops);  tops.push\_back(root->key);  }  };  } |

calculating\_node.h

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | #pragma once  #include <zmq.hpp>  #include <string>  #include <vector>  #include <map>  #include <array>  #include "message\_type.h"  **namespace** mysys {  **class** **CalculatingNode** {  **public:**  CalculatingNode(**int** id, **int** base\_port);  CalculatingNode(**const** CalculatingNode& other) = **delete**;  CalculatingNode(CalculatingNode&& other) noexcept;  ~CalculatingNode() noexcept;  **void** **connect\_child**(**int** child\_id);  std::vector<**int**> ping\_children(); // returns id of unavailable child  MyMessage **get\_child\_msg**(zmq::**socket\_t**\* child);  MyMessage **get\_parent\_msg**();  **bool** **req**(zmq::**socket\_t**\* child, **const** MyMessage& msg);  **void** **reply**(**const** MyMessage& msg);  **int** id() **const**;  std::array<std::pair<**int**, zmq::**socket\_t**\*>, **2**> children() **const**;  std::vector<**int**> \_string\_to\_vector(**const** std::string& str);  std::string exec(**const** std::string& text, **const** std::string& pattern);  zmq::**socket\_t**\* get\_less\_child() **const**;  zmq::**socket\_t**\* get\_greater\_child() **const**;  **private:**  zmq::**context\_t** \_context;  zmq::**socket\_t** \_s\_parent; // like server (to parent)  std::array<std::pair<**int**, zmq::**socket\_t**\*>, **2**> \_s\_children;  **int** \_base\_port;  **int** \_id;  **void** **\_msg\_to\_string**(**const** zmq::**message\_t**& msg, std::string& str);  };  } |

calculating\_node.cpp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193 | #include "calculating\_node.h"  **using** **namespace** mysys;  std::vector<**int**> PrefixFunction(**const** std::string& s) {  **unsigned** **int** n = s.size();  std::vector<**int**> p(n);  **for** (**int** i = **1**; i < n; ++i) {  p[i] = p[i - **1**];  **while** (p[i] > **0** and s[i] != s[p[i]]) {  p[i] = p[p[i] - **1**];  }  **if** (s[i] == s[p[i]]) {  ++p[i];  }  }  **return** p;  }  std::vector<**int**> KMPWeak(**const** std::string& text, **const** std::string& pattern) {  std::vector<**int**> p = PrefixFunction(pattern);  **int** m = pattern.size();  **int** n = text.size();  **int** i = **0**;  std::vector<**int**> ans;  **if** (m > n) {  **return** ans;  }  **while** (i < n - m + **1**) {  **int** j = **0**;  **while** (j < m and pattern[j] == text[i + j]) {  ++j;  }  **if** (j == m) {  ans.push\_back(i);  } **else** {  **if** (j > **0** and j > p[j - **1**]) {  i = i + j - p[j - **1**] - **1**;  }  }  ++i;  }  **return** ans;  }  CalculatingNode::CalculatingNode(**int** id, **int** base\_port) :  \_id{id},  \_base\_port{base\_port},  \_s\_parent(\_context, zmq::socket\_type::pair) {  \_s\_parent.set(zmq::sockopt::sndtimeo, **3000**);  std::string addr = "tcp://localhost:" + std::to\_string(\_base\_port + \_id);  \_s\_parent.connect(addr);  \_s\_children[**0**] = std::make\_pair(-**1**, nullptr);  \_s\_children[**1**] = std::make\_pair(-**1**, nullptr);  }  CalculatingNode::CalculatingNode(CalculatingNode&& other) noexcept {  \_context = std::move(other.\_context);  \_s\_parent = std::move(other.\_s\_parent);  \_s\_children = std::move(other.\_s\_children);  \_base\_port = std::move(\_base\_port);  \_id = std::move(other.\_id);  }  CalculatingNode::~CalculatingNode() noexcept {  **for** (**auto**& p : \_s\_children) {  **if** (p.second == nullptr) **continue**;  **delete** p.second;  }  }  **bool** CalculatingNode::req(zmq::**socket\_t**\* child, **const** MyMessage& msg) {  zmq::**message\_t** message\_type(std::to\_string(msg.type));  **if** (msg.type == MessageType::ping) {  **auto** res = child->send(message\_type, zmq::send\_flags::dontwait);  **if** (!res) {  **return** false;  }  **return** true;  }  **auto** res = child->send(message\_type, zmq::send\_flags::sndmore);  zmq::**message\_t** message\_text(msg.text);  res = child->send(message\_text, zmq::send\_flags::none);  **return** true;  }  **void** CalculatingNode::reply(**const** MyMessage& msg) {  zmq::**message\_t** message\_type(std::to\_string((**int**) msg.type));  **auto** res = \_s\_parent.send(message\_type, zmq::send\_flags::sndmore);  zmq::**message\_t** message\_text(msg.text);  res = \_s\_parent.send(message\_text, zmq::send\_flags::none);  }  **void** CalculatingNode::connect\_child(**int** child\_id) {  zmq::**socket\_t**\* child = **new** zmq::**socket\_t**(\_context, zmq::socket\_type::pair);  std::string addr = "tcp://\*:" + std::to\_string(\_base\_port + child\_id);  child->bind(addr);  **if** (child\_id < \_id) {  \_s\_children[**0**] = std::make\_pair(child\_id, child);  } **else** {  \_s\_children[**1**] = std::make\_pair(child\_id, child);  }  }  **void** CalculatingNode::\_msg\_to\_string(**const** zmq::**message\_t**& msg, std::string& str) {  str.resize(msg.size() / **sizeof**(**char**));  std::memcpy(str.data(), msg.data(), msg.size());  }  MyMessage CalculatingNode::get\_child\_msg(zmq::**socket\_t**\* child) {  MyMessage msg;  zmq::**message\_t** msg\_type;  **auto** res = child->recv(msg\_type);  std::string buf;  \_msg\_to\_string(msg\_type, buf);  msg.type = (MessageType) std::stoi(buf);  zmq::**message\_t** msg\_text;  res = child->recv(msg\_text);  \_msg\_to\_string(msg\_text, buf);  msg.text = buf;  **return** msg;  }  MyMessage CalculatingNode::get\_parent\_msg() {  MyMessage msg;  zmq::**message\_t** msg\_type;  **auto** res = \_s\_parent.recv(msg\_type);  std::string buf;  \_msg\_to\_string(msg\_type, buf);  msg.type = (MessageType) std::stoi(buf);  **if** (msg.type == MessageType::ping || msg.type == MessageType::shutdown) **return** msg;  zmq::**message\_t** msg\_text;  res = \_s\_parent.recv(msg\_text);  \_msg\_to\_string(msg\_text, buf);  msg.text = buf;  **return** msg;  }  std::vector<**int**> CalculatingNode::\_string\_to\_vector(**const** std::string& str) {  std::stringstream ss(str);  std::vector<**int**> vec;  **int** num;  **while** (ss >> num) {  vec.push\_back(num);  }  **return** vec;  }  std::vector<**int**> CalculatingNode::ping\_children() {  std::string s = "";  std::vector<**int**> ids;  **for** (**auto**& p : \_s\_children) {  **if** (p.second == nullptr) **continue**;  MyMessage msg;  msg.type = MessageType::ping;  **bool** res = req(p.second, msg);  **if** (!res) {  s += std::to\_string(p.first) + ' ';  **continue**;  }  msg = get\_child\_msg(p.second);  s += msg.text + ' ';  }  ids = \_string\_to\_vector(s);  **return** ids;  }  **int** CalculatingNode::id() **const** {  **return** \_id;  }  std::array<std::pair<**int**, zmq::**socket\_t**\*>, **2**> CalculatingNode::children() **const** {  **return** \_s\_children;  }  std::string CalculatingNode::exec(**const** std::string& text, **const** std::string& pattern) {  std::vector<**int**> idxs = KMPWeak(text, pattern);  std::string res = "";  **for** (**auto** idx : idxs) {  res += std::to\_string(idx) + ' ';  }  **return** res;  }  zmq::**socket\_t**\* CalculatingNode::get\_less\_child() **const** {  **return** \_s\_children[**0**].second;  }  zmq::**socket\_t**\* CalculatingNode::get\_greater\_child() **const** {  **return** \_s\_children[**1**].second;  } |

control\_node.h

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | #pragma once  #include <zmq.hpp>  #include <unistd.h>  #include <vector>  #include <string>  #include <fstream>  #include "message\_type.h"  #include "binary\_tree.h"  **namespace** mysys {  **class** **ControlNode** {  **public:**  ControlNode(**int** base\_port = **5000**);  ControlNode(**const** ControlNode& other) = **delete**;  ControlNode(ControlNode&& other) noexcept;  ~ControlNode() noexcept;  **pid\_t** **new\_node**(**int** id);  std::vector<**int**> pingall();  std::string exec(**int** id, **const** std::string& text, **const** std::string& pattern);  MyMessage **get\_message**(zmq::recv\_flags flags = zmq::recv\_flags::none);  **bool** **send\_message**(**const** MyMessage& msg);  **private:**  zmq::**context\_t** \_context;  zmq::**socket\_t** \_s\_request;  **int** \_base\_port;  BinaryTree<**int**> \_topology;  **bool** \_has\_child;  std::vector<**int**> \_string\_to\_vector(**const** std::string& str);  **void** **\_msg\_to\_string**(**const** zmq::**message\_t**& msg, std::string& str);  };  } |

control\_node.cpp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118 | #include "control\_node.h"  #include <iostream>  **using** **namespace** mysys;  **void** ControlNode::\_msg\_to\_string(**const** zmq::**message\_t**& msg, std::string& str) {  str.resize(msg.size() / **sizeof**(**char**));  std::memcpy(str.data(), msg.data(), msg.size());  }  **template**<**typename** Item>  **void** concat(std::vector<Item> &a, std::vector<Item> &b) {  a.reserve(a.size() + b.size());  a.insert(  a.end(),  std::make\_move\_iterator(b.begin()),  std::make\_move\_iterator(b.end())  );  }  ControlNode::ControlNode(**int** base\_port) : \_base\_port{base\_port},  \_s\_request(\_context, zmq::socket\_type::pair), \_has\_child{false} {  \_s\_request.set(zmq::sockopt::sndtimeo, **3000**);  }  ControlNode::ControlNode(ControlNode&& other) noexcept {  \_context = std::move(other.\_context);  \_s\_request = std::move(other.\_s\_request);  \_base\_port = std::move(other.\_base\_port);  \_topology = std::move(other.\_topology);  }  ControlNode::~ControlNode() noexcept {}  **pid\_t** ControlNode::new\_node(**int** id) {  **if** (id == **0**) **throw** std::logic\_error("id 0 is reserved for server");  \_topology.insert(id);  **int** parent = \_topology.get\_parent(id);  **pid\_t** pid = fork();  **if** (pid == **0**) {  execl("./lab5-7\_calc", "./lab5-7\_calc", std::to\_string(id).c\_str(), std::to\_string(\_base\_port).c\_str());  } **else** {  **if** (!\_has\_child) {  std::string addr = "tcp://\*:" + std::to\_string(\_base\_port + id);  \_s\_request.bind(addr);  \_has\_child = true;  } **else** {  MyMessage bind;  bind.type = MessageType::bind\_node;  bind.text = std::to\_string(parent) + " " + std::to\_string(id);  send\_message(bind);  }  **return** pid;  }  }  std::vector<**int**> ControlNode::\_string\_to\_vector(**const** std::string& str) {  std::stringstream ss(str);  std::vector<**int**> vec;  **int** num;  **while** (ss >> num) {  vec.push\_back(num);  }  **return** vec;  }  MyMessage ControlNode::get\_message(zmq::recv\_flags flags) {  MyMessage msg;  std::string buf;  zmq::**message\_t** rec;  **auto** res = \_s\_request.recv(rec, flags);  \_msg\_to\_string(rec, buf);  msg.type = (MessageType) std::stoi(buf);  res = \_s\_request.recv(rec);  \_msg\_to\_string(rec, buf);  msg.text = buf;  **return** msg;  }  **bool** ControlNode::send\_message(**const** MyMessage& msg) {  zmq::**message\_t** msg\_type(std::to\_string(msg.type));  **if** (msg.type == MessageType::ping || msg.type == MessageType::shutdown) {  **auto** res = \_s\_request.send(msg\_type, zmq::send\_flags::dontwait);  **if** (!res) **return** false;  **return** true;  }  \_s\_request.send(msg\_type, zmq::send\_flags::sndmore);  zmq::**message\_t** msg\_text(msg.text);  \_s\_request.send(msg\_text, zmq::send\_flags::none);  **return** true;  }  std::vector<**int**> ControlNode::pingall() {  MyMessage msg;  msg.type = MessageType::ping;  **if** (!send\_message(msg)) {  std::vector<**int**> ids = \_topology.get\_tops();  **return** ids;  }  msg = get\_message();  std::vector<**int**> ids = \_string\_to\_vector(msg.text);  std::vector<**int**> tops;  **for** (**auto** el : ids) {  std::vector<**int**> children = \_topology.get\_children(el);  concat<**int**>(tops, children);  }  **return** tops;  }  std::string ControlNode::exec(**int** id, **const** std::string& text, **const** std::string& pattern) {  MyMessage msg;  msg.type = MessageType::exec;  msg.text = std::to\_string(id) + ' ' + text + ' ' + pattern;  send\_message(msg);  msg = get\_message();  **if** (msg.type != MessageType::exec\_result) **throw** std::runtime\_error("Wrong message type");  **return** msg.text;  } |

message\_type.h

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | #pragma once  #include <string>  #include <zmq.hpp>  **namespace** mysys {  **enum** MessageType {  exec,  ping,  bind\_node,  exec\_result,  ping\_result,  error,  shutdown  };  **struct** MyMessage {  MessageType type;  std::string text;  };  } |

calc\_main.cpp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73 | #include "calculating\_node.h"  **using** **namespace** mysys;  std::string vector\_to\_string(**const** std::vector<**int**>& v) {  std::string str = "";  **for** (**auto** el : v) {  str += std::to\_string(el) + ' ';  }  **return** str;  }  **int** main(**int** argc, **char**\*\* argv) {  CalculatingNode node(std::stoi(argv[**1**]), std::stoi(argv[**2**]));  **while** (true) {  MyMessage msg = node.get\_parent\_msg();  **if** (msg.type == MessageType::ping) {  std::vector<**int**> ids = node.ping\_children();  MyMessage rep;  rep.type = MessageType::ping\_result;  rep.text = vector\_to\_string(ids);  node.reply(rep);  } **else** **if** (msg.type == MessageType::bind\_node) {  **int** parent, id;  std::stringstream ss(msg.text);  ss >> parent >> id;  **if** (parent == node.id()) {  node.connect\_child(id);  } **else** **if** (parent < node.id()) {  node.req(node.children()[**0**].second, msg);  } **else** {  node.req(node.children()[**1**].second, msg);  }  } **else** **if** (msg.type == MessageType::exec) {  **int** id;  std::stringstream ss(msg.text);  ss >> id;  **if** (id == node.id()) {  std::string text, pattern;  ss >> text >> pattern;  MyMessage reply;  reply.text = node.exec(text, pattern);  reply.type = MessageType::exec\_result;  node.reply(reply);  } **else** {  MyMessage next;  next.type = MessageType::exec;  next.text = msg.text;  **if** (id < node.id()) {  node.req(node.get\_less\_child(), next);  next = node.get\_child\_msg(node.get\_less\_child());  } **else** {  node.req(node.get\_greater\_child(), next);  next = node.get\_child\_msg(node.get\_greater\_child());  }  node.reply(next);  }  } **else** **if** (msg.type == MessageType::exec\_result) {  MyMessage next;  next.type = MessageType::exec\_result;  next.text = msg.text;  node.reply(next);  } **else** **if** (msg.type == MessageType::shutdown) {  MyMessage next;  next.type = MessageType::shutdown;  **if** (node.get\_less\_child() != nullptr) node.req(node.get\_less\_child(), next);  **if** (node.get\_greater\_child() != nullptr) node.req(node.get\_greater\_child(), next);  **break**;  }  }  **return** **0**;  } |

main.cpp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63 | #include <iostream>  #include <sstream>  #include "control\_node.h"  **using** **namespace** mysys;  std::vector<**int**> string\_to\_vector(**const** std::string& str) {  std::stringstream ss(str);  std::vector<**int**> vec;  **int** num;  **while** (ss >> num) {  vec.push\_back(num);  }  **return** vec;  }  **int** main() {  ControlNode ctrl;  std::cout << "Starting Control Node" << std::endl;  std::string cmd;  **while**(true) {  std::cout << "> ";  std::cin >> cmd;  **if** (cmd == "create") {  **int** id;  std::cin >> id;  **pid\_t** pid;  try {  pid = ctrl.new\_node(id);  } **catch** (std::exception& e) {  std::cout << e.what() << std::endl;  **continue**;  }  std::cout << "Ok: " << pid << std::endl;  } **else** **if** (cmd == "pingall") {  std::vector<**int**> ids = ctrl.pingall();  std::cout << "Ok: ";  **if** (ids.size() == **0**) {  std::cout << "-1";  }  **for** (**auto** id : ids) {  std::cout << id << ";";  }  std::cout << std::endl;  } **else** **if** (cmd == "q") {  MyMessage msg;  msg.type = MessageType::shutdown;  ctrl.send\_message(msg);  **break**;  } **else** **if** (cmd == "exec") {  **int** id;  std::cin >> id;  std::string text\_str, pattern\_str;  std::cin >> text\_str >> pattern\_str;  std::string res = ctrl.exec(id, text\_str, pattern\_str);  std::cout << res << std::endl;  } **else** {  std::cout << "Wrong command!" << std::endl;  }  }  **return** **0**;  } |

**Протокол работы программы**

**Strace:**

execve("./lab5-7\_exe", ["./lab5-7\_exe"], 0x7ffda5c591e0 /\* 27 vars \*/) = 0

brk(NULL) = 0x562f286dd000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffe5d1aae50) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3ded59b000

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

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=32515, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 32515, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f3ded593000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=583920, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 585912, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f3ded503000

mmap(0x7f3ded51b000, 364544, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18000) = 0x7f3ded51b000

mmap(0x7f3ded574000, 90112, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x71000) = 0x7f3ded574000

mmap(0x7f3ded58a000, 36864, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x86000) = 0x7f3ded58a000

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) = 0x7f3ded2d7000

mprotect(0x7f3ded371000, 1576960, PROT\_NONE) = 0

mmap(0x7f3ded371000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7f3ded371000

mmap(0x7f3ded482000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7f3ded482000

mmap(0x7f3ded4f2000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7f3ded4f2000

mmap(0x7f3ded500000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f3ded500000

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) = 0x7f3ded2b7000

mmap(0x7f3ded2ba000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f3ded2ba000

mmap(0x7f3ded2d1000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7f3ded2d1000

mmap(0x7f3ded2d5000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7f3ded2d5000

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 =\340\2563\265?\356\25x\261\27\313A#\350"..., 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) = 0x7f3ded08f000

mmap(0x7f3ded0b7000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7f3ded0b7000

mmap(0x7f3ded24c000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7f3ded24c000

mmap(0x7f3ded2a4000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x214000) = 0x7f3ded2a4000

mmap(0x7f3ded2aa000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f3ded2aa000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libunwind.so.8", 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=63744, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 109264, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f3ded074000

mmap(0x7f3ded076000, 40960, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f3ded076000

mmap(0x7f3ded080000, 12288, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xc000) = 0x7f3ded080000

mmap(0x7f3ded083000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7f3ded083000

mmap(0x7f3ded085000, 39632, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f3ded085000

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) = 0x7f3ded01c000

mprotect(0x7f3ded028000, 303104, PROT\_NONE) = 0

mmap(0x7f3ded028000, 229376, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xc000) = 0x7f3ded028000

mmap(0x7f3ded060000, 69632, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x44000) = 0x7f3ded060000

mmap(0x7f3ded072000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x55000) = 0x7f3ded072000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3ded01a000

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) = 0x7f3decfc9000

mmap(0x7f3decfcd000, 172032, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f3decfcd000

mmap(0x7f3decff7000, 118784, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2e000) = 0x7f3decff7000

mmap(0x7f3ded014000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4a000) = 0x7f3ded014000

mmap(0x7f3ded016000, 14416, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f3ded016000

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) = 0x7f3dece9e000

mprotect(0x7f3decea8000, 446464, PROT\_NONE) = 0

mmap(0x7f3decea8000, 286720, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa000) = 0x7f3decea8000

mmap(0x7f3deceee000, 155648, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x50000) = 0x7f3deceee000

mmap(0x7f3decf15000, 16384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x76000) = 0x7f3decf15000

mmap(0x7f3decf19000, 719360, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f3decf19000

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) = 0x7f3dece4a000

mprotect(0x7f3dece55000, 282624, PROT\_NONE) = 0

mmap(0x7f3dece55000, 229376, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb000) = 0x7f3dece55000

mmap(0x7f3dece8d000, 49152, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x43000) = 0x7f3dece8d000

mmap(0x7f3dece9a000, 16384, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4f000) = 0x7f3dece9a000

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) = 0x7f3decd63000

mmap(0x7f3decd71000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7f3decd71000

mmap(0x7f3decded000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7f3decded000

mmap(0x7f3dece48000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7f3dece48000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/liblzma.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\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=170456, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 172296, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f3decd38000

mmap(0x7f3decd3b000, 110592, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f3decd3b000

mmap(0x7f3decd56000, 45056, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e000) = 0x7f3decd56000

mmap(0x7f3decd61000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7f3decd61000

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) = 0x7f3decd36000

mmap(NULL, 16424, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f3decd31000

mmap(0x7f3decd32000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7f3decd32000

mmap(0x7f3decd33000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f3decd33000

mmap(0x7f3decd34000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f3decd34000

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) = 0x7f3decc66000

mprotect(0x7f3decc87000, 634880, PROT\_NONE) = 0

mmap(0x7f3decc87000, 380928, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21000) = 0x7f3decc87000

mmap(0x7f3decce4000, 249856, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x7e000) = 0x7f3decce4000

mmap(0x7f3decd22000, 61440, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xbb000) = 0x7f3decd22000

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) = 0x7f3decc37000

mprotect(0x7f3decc3b000, 163840, PROT\_NONE) = 0

mmap(0x7f3decc3b000, 110592, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f3decc3b000

mmap(0x7f3decc56000, 49152, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1f000) = 0x7f3decc56000

mmap(0x7f3decc63000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2b000) = 0x7f3decc63000

mmap(0x7f3decc65000, 56, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f3decc65000

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) = 0x7f3decc31000

mmap(0x7f3decc33000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f3decc33000

mmap(0x7f3decc34000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f3decc34000

mmap(0x7f3decc35000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f3decc35000

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) = 0x7f3decc23000

mprotect(0x7f3decc26000, 36864, PROT\_NONE) = 0

mmap(0x7f3decc26000, 24576, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f3decc26000

mmap(0x7f3decc2c000, 8192, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9000) = 0x7f3decc2c000

mmap(0x7f3decc2f000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb000) = 0x7f3decc2f000

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) = 0x7f3decc21000

mmap(NULL, 24592, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f3decc1a000

mmap(0x7f3decc1c000, 8192, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7f3decc1c000

mmap(0x7f3decc1e000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f3decc1e000

mmap(0x7f3decc1f000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f3decc1f000

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) = 0x7f3decc06000

mmap(0x7f3decc09000, 40960, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f3decc09000

mmap(0x7f3decc13000, 12288, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xd000) = 0x7f3decc13000

mmap(0x7f3decc16000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xf000) = 0x7f3decc16000

mmap(0x7f3decc18000, 6728, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f3decc18000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3decc04000

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3decc01000

arch\_prctl(ARCH\_SET\_FS, 0x7f3decc019c0) = 0

set\_tid\_address(0x7f3decc01c90) = 13203

set\_robust\_list(0x7f3decc01ca0, 24) = 0

rseq(0x7f3decc02360, 0x20, 0, 0x53053053) = 0

mprotect(0x7f3ded2a4000, 16384, PROT\_READ) = 0

mprotect(0x7f3decc16000, 4096, PROT\_READ) = 0

mprotect(0x7f3decc1f000, 4096, PROT\_READ) = 0

mprotect(0x7f3decc2f000, 4096, PROT\_READ) = 0

mprotect(0x7f3decc35000, 4096, PROT\_READ) = 0

mprotect(0x7f3decc63000, 4096, PROT\_READ) = 0

mprotect(0x7f3decd22000, 53248, PROT\_READ) = 0

mprotect(0x7f3decd34000, 4096, PROT\_READ) = 0

mprotect(0x7f3decd61000, 4096, PROT\_READ) = 0

mprotect(0x7f3dece48000, 4096, PROT\_READ) = 0

mprotect(0x7f3dece9a000, 8192, PROT\_READ) = 0

mprotect(0x7f3ded2d5000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3decbff000

mprotect(0x7f3ded4f2000, 45056, PROT\_READ) = 0

mprotect(0x7f3decf15000, 12288, PROT\_READ) = 0

mprotect(0x7f3ded014000, 4096, PROT\_READ) = 0

mprotect(0x7f3ded072000, 4096, PROT\_READ) = 0

mprotect(0x7f3ded083000, 4096, PROT\_READ) = 0

mprotect(0x7f3ded58a000, 32768, PROT\_READ) = 0

mprotect(0x562f26c92000, 4096, PROT\_READ) = 0

mprotect(0x7f3ded5d5000, 8192, PROT\_READ) = 0

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

munmap(0x7f3ded593000, 32515) = 0

getrandom("\xc7\x7b\x4f\x88\xaa\xce\xa4\xba", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x562f286dd000

brk(0x562f286fe000) = 0x562f286fe000

futex(0x7f3ded50077c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

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

read(3, "0-15\n", 1024) = 5

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, 0x562f286eeee0 /\* 31 entries \*/, 32768) = 896

getdents64(3, 0x562f286eeee0 /\* 0 entries \*/, 32768) = 0

close(3) = 0

getpid() = 13203

sched\_getaffinity(13203, 128, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]) = 32

newfstatat(AT\_FDCWD, "/etc/nsswitch.conf", {st\_mode=S\_IFREG|0644, st\_size=510, ...}, 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=510, ...}, AT\_EMPTY\_PATH) = 0

read(3, "# /etc/nsswitch.conf\n#\n# Example"..., 4096) = 510

read(3, "", 4096) = 0

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=510, ...}, 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=32515, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 32515, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f3ded593000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/glibc-hwcaps/x86-64-v3/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/glibc-hwcaps/x86-64-v3", 0x7ffe5d1a7e50, 0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/glibc-hwcaps/x86-64-v2/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/glibc-hwcaps/x86-64-v2", 0x7ffe5d1a7e50, 0) = -1 ENOENT (No such file or directory)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls/x86\_64/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/tls", 0x7ffe5d1a7e50, 0) = -1 ENOENT (No such file or directory)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/x86\_64/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/lib/x86\_64-linux-gnu", {st\_mode=S\_IFDIR|0755, st\_size=36864, ...}, 0) = 0

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/glibc-hwcaps/x86-64-v3/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/glibc-hwcaps/x86-64-v3", 0x7ffe5d1a7e50, 0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/glibc-hwcaps/x86-64-v2/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/glibc-hwcaps/x86-64-v2", 0x7ffe5d1a7e50, 0) = -1 ENOENT (No such file or directory)

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls/x86\_64/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/tls", 0x7ffe5d1a7e50, 0) = -1 ENOENT (No such file or directory)

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/x86\_64/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu", {st\_mode=S\_IFDIR|0755, st\_size=36864, ...}, 0) = 0

openat(AT\_FDCWD, "/lib/glibc-hwcaps/x86-64-v3/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT\_FDCWD, "/lib/glibc-hwcaps/x86-64-v3", 0x7ffe5d1a7e50, 0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/lib/glibc-hwcaps/x86-64-v2/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT\_FDCWD, "/lib/glibc-hwcaps/x86-64-v2", 0x7ffe5d1a7e50, 0) = -1 ENOENT (No such file or directory)

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

newfstatat(AT\_FDCWD, "/lib/tls/x86\_64/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/lib/tls/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/lib/tls/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/lib/tls", 0x7ffe5d1a7e50, 0) = -1 ENOENT (No such file or directory)

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

newfstatat(AT\_FDCWD, "/lib/x86\_64/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/lib/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/lib/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/lib", {st\_mode=S\_IFDIR|0755, st\_size=12288, ...}, 0) = 0

openat(AT\_FDCWD, "/usr/lib/glibc-hwcaps/x86-64-v3/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT\_FDCWD, "/usr/lib/glibc-hwcaps/x86-64-v3", 0x7ffe5d1a7e50, 0) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/usr/lib/glibc-hwcaps/x86-64-v2/libnss\_db.so.2", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT\_FDCWD, "/usr/lib/glibc-hwcaps/x86-64-v2", 0x7ffe5d1a7e50, 0) = -1 ENOENT (No such file or directory)

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

newfstatat(AT\_FDCWD, "/usr/lib/tls/x86\_64/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/usr/lib/tls/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/usr/lib/tls/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/usr/lib/tls", 0x7ffe5d1a7e50, 0) = -1 ENOENT (No such file or directory)

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

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/usr/lib/x86\_64", 0x7ffe5d1a7e50, 0) = -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)

newfstatat(AT\_FDCWD, "/usr/lib", {st\_mode=S\_IFDIR|0755, st\_size=12288, ...}, 0) = 0

munmap(0x7f3ded593000, 32515) = 0

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=32515, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 32515, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f3ded593000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libnss\_db-2.35.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/usr/lib/x86\_64-linux-gnu/libnss\_db-2.35.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/lib/libnss\_db-2.35.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/usr/lib/libnss\_db-2.35.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory)

munmap(0x7f3ded593000, 32515) = 0

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=2932, ...}, AT\_EMPTY\_PATH) = 0

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

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

read(3, "", 4096) = 0

close(3) = 0

eventfd2(0, EFD\_CLOEXEC) = 3

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

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

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

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

getpid() = 13203

getpid() = 13203

getrandom("\x6c\x5c\xe2\x32\x30\xe6\xf2\x1b\xdc\x2d\xac\xd8\x40\xe0\x20\x73", 16, 0) = 16

getrandom("\xfb\x4c\x41\xdf\x15\xf6\x1d\x18\x9b\x44\x17\x2f\xd9\x93\xd8\x6b", 16, 0) = 16

eventfd2(0, EFD\_CLOEXEC) = 4

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

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

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

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

getpid() = 13203

epoll\_create1(EPOLL\_CLOEXEC) = 5

epoll\_ctl(5, EPOLL\_CTL\_ADD, 4, {events=0, data={u32=678359648, u64=94760541811296}}) = 0

epoll\_ctl(5, EPOLL\_CTL\_MOD, 4, {events=EPOLLIN, data={u32=678359648, u64=94760541811296}}) = 0

getpid() = 13203

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7f3ded120870, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_ONSTACK|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7f3ded0d1520}, NULL, 8) = 0

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

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f3dec3fe000

mprotect(0x7f3dec3ff000, 8388608, PROT\_READ|PROT\_WRITE) = 0

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

clone3({flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, child\_tid=0x7f3decbfe910, parent\_tid=0x7f3decbfe910, exit\_signal=0, stack=0x7f3dec3fe000, stack\_size=0x7ffc80, tls=0x7f3decbfe640} => {parent\_tid=[13204]}, 88) = 13204

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

eventfd2(0, EFD\_CLOEXEC) = 6

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

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

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

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

getpid() = 13203

epoll\_create1(EPOLL\_CLOEXEC) = 7

epoll\_ctl(7, EPOLL\_CTL\_ADD, 6, {events=0, data={u32=678363568, u64=94760541815216}}) = 0

epoll\_ctl(7, EPOLL\_CTL\_MOD, 6, {events=EPOLLIN, data={u32=678363568, u64=94760541815216}}) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f3debbfd000

mprotect(0x7f3debbfe000, 8388608, PROT\_READ|PROT\_WRITE) = 0

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

clone3({flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, child\_tid=0x7f3dec3fd910, parent\_tid=0x7f3dec3fd910, exit\_signal=0, stack=0x7f3debbfd000, stack\_size=0x7ffc80, tls=0x7f3dec3fd640} => {parent\_tid=[13205]}, 88) = 13205

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

eventfd2(0, EFD\_CLOEXEC) = 8

fcntl(8, F\_GETFL) = 0x2 (flags O\_RDWR)

fcntl(8, F\_SETFL, O\_RDWR|O\_NONBLOCK) = 0

fcntl(8, F\_GETFL) = 0x802 (flags O\_RDWR|O\_NONBLOCK)

fcntl(8, F\_SETFL, O\_RDWR|O\_NONBLOCK) = 0

getpid() = 13203

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

write(1, "Starting Control Node\n", 22Starting Control Node

) = 22

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

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

read(0, create 1

"create 1\n", 1024) = 9

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f3decc01c90) = 13218

getpid() = 13203

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

**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(5001), sin\_addr=inet\_addr("0.0.0.0")}, 16) = 0**

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

**getsockname(9, {sa\_family=AF\_INET, sin\_port=htons(5001), sin\_addr=inet\_addr("0.0.0.0")}, [128 => 16]) = 0**

**getsockname(9, {sa\_family=AF\_INET, sin\_port=htons(5001), sin\_addr=inet\_addr("0.0.0.0")}, [128 => 16]) = 0**

getpid() = 13203

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

getpid() = 13203

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

write(1, "Ok: 13218\n", 10Ok: 13218

) = 10

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

read(0, pingall

"pingall\n", 1024) = 8

getpid() = 13203

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

getpid() = 13203

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

getpid() = 13203

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

getpid() = 13203

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

getpid() = 13203

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

getpid() = 13203

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

getpid() = 13203

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

getpid() = 13203

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

write(1, "Ok: -1\n", 7Ok: -1

) = 7

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

read(0, q

"q\n", 1024) = 2

getpid() = 13203

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

getpid() = 13203

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

getpid() = 13203

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

getpid() = 13203

getpid() = 13203

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

futex(0x562f286ef388, FUTEX\_WAKE\_PRIVATE, 1) = 1

getpid() = 13203

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=13218, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

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

getpid() = 13203

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

getpid() = 13203

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

close(7) = 0

close(6) = 0

close(5) = 0

close(4) = 0

close(3) = 0

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

exit\_group(0) = ?

+++ exited with 0 +++

**Тестирование:**

cat\_mood@nuclear-box:~/programming/mai-os-labs/lab5-7/build$ ./lab5-7\_exe

Starting Control Node

> create 4

Ok: 7751

> create 2

Ok: 7784

> create 1

Ok: 7787

> create 3

Ok: 7802

>

pingall

Ok: -1

> create 3

Already exists

> pingall

Ok: -1

> exec 1 abcabcabc abc

0 3 6

> exec 4 abcabcabc abc

0 3 6

terminal\_2> kill -9 7784

> pingall

Ok: 1;3;2;

> q

**Вывод**

В ходе лабораторной работы я получил опыт разработки распределённой системы. Научился работать с сокетами и очередями сообщений.