Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторные работы №6-8 по курсу «Операционные системы»

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

Студент: Аминов Степан Сергеевич
Группа: М80 – 308Б-19
Вариант: 11
Преподаватель: Миронов Евгений Сергеевич
Дата: 09.06.2021
Оценка:
Подпись: ___________

Москва, 2021

1. Постановка задачи

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

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

2. Общие сведения о программе

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

Программа поддерживает следующие команды:

- create [id] [parent_id] – создать новый узел [id], родителем которого является узел [parent_id]. Если [parent_id] = -1, то родительский узел – управляющий.
- kill [id] – удалить узел [id]. Все дочерние узлы будут также удалены.
- ping – проверить доступность узла.
- exit – выйти из программы.

3. Общий метод и алгоритм решения

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

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

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

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

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

Для хранения локального словаря используется контейнер std::unordered_map. Для проверки доступности узлов используется контейнер std::set. Управляющий узел отправляет запрос всем спискам узлов и получает в ответ строку с id всех доступных узлов списка. Все id добавляются в set, а потом выводятся на экран.

4. Основные файлы программы

topology.h

```
#include <list>
#include <stdexcept>

class topology {
private:
   std::list<std::list<int>> container;

public:
   void insert(int id, int parent_id) {
      if (parent_id == -1) {
         std::list<int> new_list;
         new_list.push_back(id);
         container.push_back(new_list);
      }
      else {
         int list_id = find(parent_id);
         if (list_id == -1) {
            throw std::runtime_error("Wrong parent id");
         }
         auto it1 = container.begin();
         std::advance(it1, list_id);
         for (auto it2 = it1->begin(); it2 != it1->end(); ++it2) {
            if (*it2 == parent_id) {
               it1->insert(++it2, id);
               return;
            }
         }
      }
   }

   int find(int id) {
      int cur_list_id = 0;
      for (auto it1 = container.begin(); it1 != container.end(); ++it1) {
         for (auto it2 = it1->begin(); it2 != it1->end(); ++it2) {
            if (*it2 == id) {
               return cur_list_id;
            }
         }
         ++cur_list_id;
      }
      return -1;
   }

   void erase(int id) {
      int list_id = find(id);
      if (list_id == -1) {
         throw std::runtime_error("Wrong id");
```

```
        }
        auto it1 = container.begin();
        std::advance(it1, list_id);
        for (auto it2 = it1->begin(); it2 != it1->end(); ++it2) {
            if (*it2 == id) {
                it1->erase(it2, it1->end());
                if (it1->empty()) {
                    container.erase(it1);
                }
                return;
            }
        }
    }

    int get_first_id(int list_id) {
        auto it1 = container.begin();
        std::advance(it1, list_id);
        if (it1->begin() == it1->end()) {
            return -1;
        }
        return *(it1->begin());
    }
};
```

zmq.h

```
#include <zmq.hpp>
#include <iostream>

const int MAIN_PORT = 4040;

void send_message(zmq::socket_t& socket, const std::string& msg) {
    zmq::message_t message(msg.size());
    memcpy(message.data(), msg.c_str(), msg.size());
    socket.send(message);
}

std::string receive_message(zmq::socket_t& socket) {
    zmq::message_t message;
    int chars_read;
    try {
        chars_read = (int)socket.recv(&message);
    }
    catch (...) {
        chars_read = 0;
    }
    if (chars_read == 0) {
        return "Error: node is unavailable [zmq_func]";
    }
    std::string received_msg(static_cast<char*>(message.data()), message.size());
    return received_msg;
}

void connect(zmq::socket_t& socket, int id) {
    std::string address = "tcp://127.0.0.1:" + std::to_string(MAIN_PORT + id);
    socket.connect(address);
}

void disconnect(zmq::socket_t& socket, int id) {
    std::string address = "tcp://127.0.0.1:" + std::to_string(MAIN_PORT + id);
    socket.disconnect(address);
}

void bind(zmq::socket_t& socket, int id) {
    std::string address = "tcp://127.0.0.1:" + std::to_string(MAIN_PORT + id);
    socket.bind(address);
}

void unbind(zmq::socket_t& socket, int id) {
    std::string address = "tcp://127.0.0.1:" + std::to_string(MAIN_PORT + id);
    socket.unbind(address);

}
```

main.cpp

```
#include <unistd.h>
#include <sstream>
#include <set>

#include "zmq.h"
#include "topology.h"

int main() {
    topology network;
    std::vector<zmq::socket_t> branches;
    zmq::context_t context;


    std::string cmd;
    while (std::cin >> cmd) {

        if (cmd == "create") {
            int node_id, parent_id;
            std::cin >> node_id >> parent_id;

            if (network.find(node_id) != -1) {
                std::cout << "Error: already exists" << std::endl;
            }
            else if (parent_id == -1) {
                pid_t pid = fork();
                if (pid < 0) {
                    perror("Can't create new process");
                    return -1;
                }
                if (pid == 0) {
                    execl("./node", "./node", std::to_string(node_id).c_str(), NULL);
                    perror("Can't execute new process");
                    return -2;
                }

                branches.emplace_back(context, ZMQ_REQ);
                branches[branches.size() - 1].setsockopt(ZMQ_SNDTIMEO, 5000);
                bind(branches[branches.size() - 1], node_id);
                send_message(branches[branches.size() - 1], std::to_string(node_id) + "pid");

                std::string reply = receive_message(branches[branches.size() - 1]);
                std::cout << reply << std::endl;
                network.insert(node_id, parent_id);
            }
            else if (network.find(parent_id) == -1) {
                std::cout << "Error: parent not found" << std::endl;
            }
            else {
                int branch = network.find(parent_id);
                send_message(branches[branch], std::to_string(parent_id) + "create " +
std::to_string(node_id));

                std::string reply = receive_message(branches[branch]);
                std::cout << reply << std::endl;
                network.insert(node_id, parent_id);
            }
        }
        else if (cmd == "kill") {
            int id;
            std::cin >> id;
            int branch = network.find(id);
            if (branch == -1) {
                std::cout << "ERROR: incorrect node id" << std::endl;
            }
            else {
                bool is_first = (network.get_first_id(branch) == id);
                send_message(branches[branch], std::to_string(id) + " kill");

                std::string reply = receive_message(branches[branch]);
                std::cout << reply << std::endl;
                network.erase(id);
                if (is_first) {
                    unbind(branches[branch], id);
                    branches.erase(branches.begin() + branch);
                }
            }
        }
        else if (cmd == "ping") {
            int id;
            std::set<int> available_nodes;
            std::cin >> id;
            int branch = network.find(id);
            if (branch == -1) {
                std::cout << "ERROR: incorrect node id" << std::endl;
            }
            else {
                bool is_first = (network.get_first_id(branch) == id);
                send_message(branches[branch], std::to_string(id) + " ping");

                std::string received_message = receive_message(branches[branch]);
                std::istringstream reply(received_message);
                int node;
                while(reply >> node) {
                    available_nodes.insert(node);
                }
            }
            if (available_nodes.empty()) {
                std::cout << "Ok: 0" << std::endl;
            }
            else {
                std::cout << "Ok: 1" << std::endl;
            }
        }
        else if (cmd == "exit") {
            for (size_t i = 0; i < branches.size(); ++i) {
                int first_node_id = network.get_first_id(i);
                send_message(branches[i], std::to_string(first_node_id) + " kill");

                std::string reply = receive_message(branches[i]);
                if (reply != "OK") {
                    std::cout << reply << std::endl;
                }
                else {
                    unbind(branches[i], first_node_id);
                }
            }
            exit(0);
        }
        else {
            std::cout << "Incorrect cmd" << std::endl;
```

```
        }
    }
}
```

node.cpp

```
#include <unordered_map>
#include <unistd.h>
#include <sstream>
#include <unordered_map>

#include "zmq.h"

int main(int argc, char* argv[]) {
    if (argc != 2 && argc != 3) {
        throw std::runtime_error("Wrong args for counting node");
    }
    int cur_id = std::atoi(argv[1]);
    int child_id = -1;
    if (argc == 3) {
        child_id = std::atoi(argv[2]);
    }

    std::unordered_map<std::string, int> dictionary;

    zmq::context_t context;
    zmq::socket_t parent_socket(context, ZMQ_REP);
    connect(parent_socket, cur_id);

    zmq::socket_t child_socket(context, ZMQ_REQ);
    child_socket.setsockopt(ZMQ_SNDTIMEO, 5000);
    if (child_id != -1) {
        bind(child_socket, child_id);
    }

    std::string message;
    while (true) {
        message = receive_message(parent_socket);
        std::istringstream request(message);
        int dest_id;
        request >> dest_id;

        std::string cmd;
        request >> cmd;

        if (dest_id == cur_id) {

            if (cmd == "pid") {
                send_message(parent_socket, "OK: " + std::to_string(getpid()));
            }

            else if (cmd == "create") {
                int new_child_id;
                request >> new_child_id;
                if (child_id != -1) {
                    unbind(child_socket, child_id);
                }
                bind(child_socket, new_child_id);
                pid_t pid = fork();
                if (pid < 0) {
                    perror("Can't create new process");
                    return -1;
                }
                if (pid == 0) {
                    execl("./node", "./node", std::to_string(new_child_id).c_str(),
std::to_string(child_id).c_str(), NULL);
                    perror("Can't execute new process");
                    return -2;
                }
                send_message(child_socket, std::to_string(new_child_id) + "pid");
                child_id = new_child_id;
                send_message(parent_socket, receive_message(child_socket));
            }

            else if (cmd == "ping") {
                std::string reply;
                if (child_id != -1) {
                    send_message(child_socket, std::to_string(child_id) + " pini    "
                                                             "");
                    std::string msg = receive_message(child_socket);
                    reply += " " + msg;
                }
                send_message(parent_socket, std::to_string(cur_id) + reply);
            }


            else if (cmd == "kill") {
                if (child_id != -1) {
                    send_message(child_socket, std::to_string(child_id) + " kill");
                    std::string msg = receive_message(child_socket);
                    if (msg == "OK") {
                        send_message(parent_socket, "OK");
                    }
                    unbind(child_socket, child_id);
                    disconnect(parent_socket, cur_id);
                    break;
                }
                send_message(parent_socket, "OK");
```

```
                disconnect(parent_socket, cur_id);
                break;
            }
        }
        else if (child_id != -1) {
            send_message(child_socket, message);
            send_message(parent_socket, receive_message(child_socket));
            if (child_id == dest_id && cmd == "kill") {
                child_id = -1;
            }
        }
        else {
            send_message(parent_socket, "Error: node is unavailable");
        }
    }
}
```

5. Демонстрация работы программы

```
magic@magical:~/CLionProjects/osserver$ ./main
create 100 -1
OK: 8681
create 3 100
OK: 8686
create 4 3
OK: 8690
create 5 4
OK: 8694
ping 5
Ok: 1
ping 4
Ok: 1
ping 3
Ok: 1
kill 3
OK
ping 5
ERROR: incorrect node id
Ok: 0
ping 4
ERROR: incorrect node id
Ok: 0
exit
magic@magical:~/CLionProjects/osserver$ strace -o strace.txt ./main
create 100 -1
OK: 8710
exit
magic@magical:~/CLionProjects/osserver$ cat strace.txt
execve("./main", ["./main"], 0x7ffc38edbdd0 /* 66 vars */) = 0
brk(NULL)                               = 0x5628e8608000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fffcc3e23f0) = -1 EINVAL (Invalid argument)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=102741, ...}) = 0
mmap(NULL, 102741, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f76e2f69000
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libzmq.so.5", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`z\1\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=675776, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f76e2f67000
mmap(NULL, 678128, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e2ec1000
mmap(0x7f76e2ed7000, 430080, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x16000) = 0x7f76e2ed7000
mmap(0x7f76e2f40000, 126976, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x7f000) = 0x7f76e2f40000
mmap(0x7f76e2f5f000, 32768, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x9d000) = 0x7f76e2f5f000
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`\341\t\0\0\0\0\0"..., 832) =
832
fstat(3, {st_mode=S_IFREG|0644, st_size=1956992, ...}) = 0
mmap(NULL, 1972224, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e2cdf000
mprotect(0x7f76e2d75000, 1290240, PROT_NONE) = 0
mmap(0x7f76e2d75000, 987136, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x96000) = 0x7f76e2d75000
mmap(0x7f76e2e66000, 299008, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x187000) = 0x7f76e2e66000
mmap(0x7f76e2eb0000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1d0000) = 0x7f76e2eb0000
mmap(0x7f76e2ebe000, 10240, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f76e2ebe000
close(3)                                = 0
```

```
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\3405\0\0\0\0\0\0"..., 832) =
832
fstat(3, {st_mode=S_IFREG|0644, st_size=104984, ...}) = 0
mmap(NULL, 107592, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e2cc4000
mmap(0x7f76e2cc7000, 73728, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x3000) = 0x7f76e2cc7000
mmap(0x7f76e2cd9000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x15000) = 0x7f76e2cd9000
mmap(0x7f76e2cdd000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x18000) = 0x7f76e2cdd000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"..., 832) =
832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784,
64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32,
848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\
274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784,
64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32,
848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\
274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e2ad2000
mprotect(0x7f76e2af7000, 1847296, PROT_NONE) = 0
mmap(0x7f76e2af7000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x25000) = 0x7f76e2af7000
mmap(0x7f76e2c6f000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x19d000) = 0x7f76e2c6f000
mmap(0x7f76e2cba000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1e7000) = 0x7f76e2cba000
mmap(0x7f76e2cc0000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f76e2cc0000
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libsodium.so.23", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\200\302\0\0\0\0\0\0"..., 832)
= 832
fstat(3, {st_mode=S_IFREG|0644, st_size=355016, ...}) = 0
mmap(NULL, 357384, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e2a7a000
mmap(0x7f76e2a86000, 229376, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xc000) = 0x7f76e2a86000
mmap(0x7f76e2abe000, 73728, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x44000) = 0x7f76e2abe000
mmap(0x7f76e2ad0000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x55000) = 0x7f76e2ad0000
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libpgm-5.2.so.0", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\240L\0\0\0\0\0\0"..., 832) =
832
fstat(3, {st_mode=S_IFREG|0644, st_size=302056, ...}) = 0
mmap(NULL, 321584, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e2a2b000
mmap(0x7f76e2a2f000, 163840, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x4000) = 0x7f76e2a2f000
mmap(0x7f76e2a57000, 118784, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x2c000) = 0x7f76e2a57000
mmap(0x7f76e2a74000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x48000) = 0x7f76e2a74000
mmap(0x7f76e2a76000, 14384, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f76e2a76000
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libnorm.so.1", O_RDONLY|O_CLOEXEC) = 3
```

```
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\257\0\0\0\0\0\0"..., 832) =
832
fstat(3, {st_mode=S_IFREG|0644, st_size=690344, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f76e2a29000
mmap(NULL, 1420000, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e28ce000
mmap(0x7f76e28d8000, 421888, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xa000) = 0x7f76e28d8000
mmap(0x7f76e293f000, 217088, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x71000) = 0x7f76e293f000
mmap(0x7f76e2974000, 16384, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xa5000) = 0x7f76e2974000
mmap(0x7f76e2978000, 723680, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7f76e2978000
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libgssapi_krb5.so.2", O_RDONLY|O_CLOEXEC)
= 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\321\0\0\0\0\0\0"..., 832) =
832
fstat(3, {st_mode=S_IFREG|0644, st_size=309712, ...}) = 0
mmap(NULL, 312128, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e2881000
mmap(0x7f76e288c000, 204800, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xb000) = 0x7f76e288c000
mmap(0x7f76e28be000, 49152, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x3d000) = 0x7f76e28be000
mmap(0x7f76e28ca000, 16384, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x48000) = 0x7f76e28ca000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\201\0\0\0\0\0\0"..., 832)
= 832
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\
223\337"..., 68, 824) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=157224, ...}) = 0
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\
223\337"..., 68, 824) = 68
mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e285e000
mmap(0x7f76e2865000, 69632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x7000) = 0x7f76e2865000
mmap(0x7f76e2876000, 20480, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x18000) = 0x7f76e2876000
mmap(0x7f76e287b000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1c000) = 0x7f76e287b000
mmap(0x7f76e287d000, 13432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f76e287d000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300\363\0\0\0\0\0\0"..., 832)
= 832
fstat(3, {st_mode=S_IFREG|0644, st_size=1369352, ...}) = 0
mmap(NULL, 1368336, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e270f000
mmap(0x7f76e271e000, 684032, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xf000) = 0x7f76e271e000
mmap(0x7f76e27c5000, 618496, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0xb6000) = 0x7f76e27c5000
mmap(0x7f76e285c000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x14c000) = 0x7f76e285c000
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libkrb5.so.3", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 ?\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=902016, ...}) = 0
mmap(NULL, 904640, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e2632000
mprotect(0x7f76e2654000, 700416, PROT_NONE) = 0
mmap(0x7f76e2654000, 397312, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x22000) = 0x7f76e2654000
mmap(0x7f76e26b5000, 299008, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x83000) = 0x7f76e26b5000
```

```
mmap(0x7f76e26ff000, 65536, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xcc000) = 0x7f76e26ff000
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libk5crypto.so.3", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\240D\0\0\0\0\0\0"..., 832) =
832
fstat(3, {st_mode=S_IFREG|0644, st_size=191040, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f76e2630000
mmap(NULL, 196696, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e25ff000
mprotect(0x7f76e2603000, 172032, PROT_NONE) = 0
mmap(0x7f76e2603000, 114688, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x4000) = 0x7f76e2603000
mmap(0x7f76e261f000, 53248, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x20000) = 0x7f76e261f000
mmap(0x7f76e262d000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x2d000) = 0x7f76e262d000
mmap(0x7f76e262f000, 88, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -
1, 0) = 0x7f76e262f000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libcom_err.so.2", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\200$\0\0\0\0\0\0"..., 832) =
832
fstat(3, {st_mode=S_IFREG|0644, st_size=22600, ...}) = 0
mmap(NULL, 24744, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e25f8000
mmap(0x7f76e25fa000, 8192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x2000) = 0x7f76e25fa000
mmap(0x7f76e25fc000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000)
= 0x7f76e25fc000
mmap(0x7f76e25fd000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x4000) = 0x7f76e25fd000
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libkrb5support.so.0", O_RDONLY|O_CLOEXEC)
= 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\3605\0\0\0\0\0\0"..., 832) =
832
fstat(3, {st_mode=S_IFREG|0644, st_size=56096, ...}) = 0
mmap(NULL, 58344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e25e9000
mmap(0x7f76e25ec000, 28672, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x3000) = 0x7f76e25ec000
mmap(0x7f76e25f3000, 12288, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xa000)
= 0x7f76e25f3000
mmap(0x7f76e25f6000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xc000) = 0x7f76e25f6000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libkeyutils.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@\"\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=22600, ...}) = 0
mmap(NULL, 24592, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e25e2000
mmap(0x7f76e25e4000, 8192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x2000) = 0x7f76e25e4000
mmap(0x7f76e25e6000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000)
= 0x7f76e25e6000
mmap(0x7f76e25e7000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x4000) = 0x7f76e25e7000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libresolv.so.2", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 G\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=101320, ...}) = 0
mmap(NULL, 113280, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e25c6000
mprotect(0x7f76e25ca000, 81920, PROT_NONE) = 0
mmap(0x7f76e25ca000, 65536, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x4000) = 0x7f76e25ca000
mmap(0x7f76e25da000, 12288, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x14000) = 0x7f76e25da000
mmap(0x7f76e25de000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x17000) = 0x7f76e25de000
```

```
mmap(0x7f76e25e0000, 6784, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f76e25e0000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libdl.so.2", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 \22\0\0\0\0\0\0"..., 832) =
832
fstat(3, {st_mode=S_IFREG|0644, st_size=18816, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f76e25c4000
mmap(NULL, 20752, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e25be000
mmap(0x7f76e25bf000, 8192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1000) = 0x7f76e25bf000
mmap(0x7f76e25c1000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000)
= 0x7f76e25c1000
mmap(0x7f76e25c2000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x3000) = 0x7f76e25c2000
close(3)                                = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f76e25bc000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f76e25ba000
arch_prctl(ARCH_SET_FS, 0x7f76e25bd600) = 0
mprotect(0x7f76e2cba000, 12288, PROT_READ) = 0
mprotect(0x7f76e25c2000, 4096, PROT_READ) = 0
mprotect(0x7f76e25de000, 4096, PROT_READ) = 0
mprotect(0x7f76e25e7000, 4096, PROT_READ) = 0
mprotect(0x7f76e25f6000, 4096, PROT_READ) = 0
mprotect(0x7f76e287b000, 4096, PROT_READ) = 0
mprotect(0x7f76e25fd000, 4096, PROT_READ) = 0
mprotect(0x7f76e262d000, 4096, PROT_READ) = 0
mprotect(0x7f76e26ff000, 57344, PROT_READ) = 0
mprotect(0x7f76e285c000, 4096, PROT_READ) = 0
mprotect(0x7f76e28ca000, 8192, PROT_READ) = 0
mprotect(0x7f76e2cdd000, 4096, PROT_READ) = 0
mprotect(0x7f76e2eb0000, 45056, PROT_READ) = 0
mprotect(0x7f76e2974000, 12288, PROT_READ) = 0
mprotect(0x7f76e2a74000, 4096, PROT_READ) = 0
mprotect(0x7f76e2ad0000, 4096, PROT_READ) = 0
mprotect(0x7f76e2f5f000, 28672, PROT_READ) = 0
mprotect(0x5628e7fe6000, 4096, PROT_READ) = 0
mprotect(0x7f76e2fb0000, 4096, PROT_READ) = 0
munmap(0x7f76e2f69000, 102741)          = 0
set_tid_address(0x7f76e25bd8d0)         = 8709
set_robust_list(0x7f76e25bd8e0, 24)     = 0
rt_sigaction(SIGRTMIN, {sa_handler=0x7f76e2865bf0, sa_mask=[], sa_flags=SA_RESTORER|
SA_SIGINFO, sa_restorer=0x7f76e28733c0}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {sa_handler=0x7f76e2865c90, sa_mask=[], sa_flags=SA_RESTORER|
SA_RESTART|SA_SIGINFO, sa_restorer=0x7f76e28733c0}, NULL, 8) = 0
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
brk(NULL)                               = 0x5628e8608000
brk(0x5628e8629000)                     = 0x5628e8629000
futex(0x7f76e2ebe6bc, FUTEX_WAKE_PRIVATE, 2147483647) = 0
futex(0x7f76e2ebe6c8, FUTEX_WAKE_PRIVATE, 2147483647) = 0
openat(AT_FDCWD, "/sys/devices/system/cpu/online", O_RDONLY|O_CLOEXEC) = 3
read(3, "0-7\n", 8192)                  = 4
close(3)                                = 0
openat(AT_FDCWD, "/sys/devices/system/cpu", O_RDONLY|O_NONBLOCK|O_CLOEXEC|O_DIRECTORY)
= 3
fstat(3, {st_mode=S_IFDIR|0755, st_size=0, ...}) = 0
getdents64(3, /* 26 entries */, 32768)  = 752
getdents64(3, /* 0 entries */, 32768)   = 0
close(3)                                = 0
getpid()                                = 8709
sched_getaffinity(8709, 128, [0, 1, 2, 3, 4, 5, 6, 7]) = 8
openat(AT_FDCWD, "/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=545, ...}) = 0
```

```
read(3, "# /etc/nsswitch.conf\n#\n# Example"..., 4096) = 545
read(3, "", 4096)                       = 0
close(3)                                = 0
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=102741, ...}) = 0
mmap(NULL, 102741, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f76e2f69000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/tls/haswell/x86_64/libnss_db.so.2", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/lib/x86_64-linux-gnu/tls/haswell/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/tls/haswell/libnss_db.so.2", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/lib/x86_64-linux-gnu/tls/haswell", 0x7fffcc3df4b0) = -1 ENOENT (No such file or
directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/tls/x86_64/libnss_db.so.2", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/lib/x86_64-linux-gnu/tls/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such file or
directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/tls/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
stat("/lib/x86_64-linux-gnu/tls", 0x7fffcc3df4b0) = -1 ENOENT (No such file or
directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/haswell/x86_64/libnss_db.so.2", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/lib/x86_64-linux-gnu/haswell/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such file
or directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/haswell/libnss_db.so.2", O_RDONLY|O_CLOEXEC) =
-1 ENOENT (No such file or directory)
stat("/lib/x86_64-linux-gnu/haswell", 0x7fffcc3df4b0) = -1 ENOENT (No such file or
directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) =
-1 ENOENT (No such file or directory)
stat("/lib/x86_64-linux-gnu/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such file or
directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
stat("/lib/x86_64-linux-gnu", {st_mode=S_IFDIR|0755, st_size=12288, ...}) = 0
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/tls/haswell/x86_64/libnss_db.so.2",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/lib/x86_64-linux-gnu/tls/haswell/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/tls/haswell/libnss_db.so.2", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/lib/x86_64-linux-gnu/tls/haswell", 0x7fffcc3df4b0) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/tls/x86_64/libnss_db.so.2", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/lib/x86_64-linux-gnu/tls/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such file
or directory)
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/tls/libnss_db.so.2", O_RDONLY|O_CLOEXEC) =
-1 ENOENT (No such file or directory)
stat("/usr/lib/x86_64-linux-gnu/tls", 0x7fffcc3df4b0) = -1 ENOENT (No such file or
directory)
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/haswell/x86_64/libnss_db.so.2", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/lib/x86_64-linux-gnu/haswell/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/haswell/libnss_db.so.2", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/lib/x86_64-linux-gnu/haswell", 0x7fffcc3df4b0) = -1 ENOENT (No such file or
directory)
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/x86_64/libnss_db.so.2", O_RDONLY|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
stat("/usr/lib/x86_64-linux-gnu/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such file or
directory)
```

```
openat(AT_FDCWD, "/usr/lib/x86_64-linux-gnu/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
stat("/usr/lib/x86_64-linux-gnu", {st_mode=S_IFDIR|0755, st_size=86016, ...}) = 0
openat(AT_FDCWD, "/lib/tls/haswell/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
stat("/lib/tls/haswell/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such file or
directory)
openat(AT_FDCWD, "/lib/tls/haswell/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
stat("/lib/tls/haswell", 0x7fffcc3df4b0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/tls/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
stat("/lib/tls/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/tls/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
stat("/lib/tls", 0x7fffcc3df4b0)        = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/haswell/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
stat("/lib/haswell/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/haswell/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
stat("/lib/haswell", 0x7fffcc3df4b0)    = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
stat("/lib/x86_64", 0x7fffcc3df4b0)     = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file
or directory)
stat("/lib", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
openat(AT_FDCWD, "/usr/lib/tls/haswell/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -
1 ENOENT (No such file or directory)
stat("/usr/lib/tls/haswell/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such file or
directory)
openat(AT_FDCWD, "/usr/lib/tls/haswell/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
stat("/usr/lib/tls/haswell", 0x7fffcc3df4b0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/lib/tls/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
stat("/usr/lib/tls/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/lib/tls/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
stat("/usr/lib/tls", 0x7fffcc3df4b0)    = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/lib/haswell/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
stat("/usr/lib/haswell/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such file or
directory)
openat(AT_FDCWD, "/usr/lib/haswell/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
stat("/usr/lib/haswell", 0x7fffcc3df4b0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/lib/x86_64/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
stat("/usr/lib/x86_64", 0x7fffcc3df4b0) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/lib/libnss_db.so.2", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
stat("/usr/lib", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
munmap(0x7f76e2f69000, 102741)          = 0
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=102741, ...}) = 0
mmap(NULL, 102741, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f76e2f69000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libnss_files.so.2", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\3005\0\0\0\0\0\0"..., 832) =
832
fstat(3, {st_mode=S_IFREG|0644, st_size=51832, ...}) = 0
mmap(NULL, 79672, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f76e25a6000
mmap(0x7f76e25a9000, 28672, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x3000) = 0x7f76e25a9000
```

```
mmap(0x7f76e25b0000, 8192, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xa000)
= 0x7f76e25b0000
mmap(0x7f76e25b2000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xb000) = 0x7f76e25b2000
mmap(0x7f76e25b4000, 22328, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f76e25b4000
close(3)                                = 0
mprotect(0x7f76e25b2000, 4096, PROT_READ) = 0
munmap(0x7f76e2f69000, 102741)          = 0
openat(AT_FDCWD, "/etc/protocols", O_RDONLY|O_CLOEXEC) = 3
lseek(3, 0, SEEK_CUR)                   = 0
fstat(3, {st_mode=S_IFREG|0644, st_size=2932, ...}) = 0
read(3, "# Internet (IP) protocols\n#\n# Up"..., 4096) = 2932
lseek(3, 0, SEEK_CUR)                   = 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
getrandom("\x76\xa5\x38\x18\xb1\x92\x68\xef\x82\x81\xdc\x5b\x37\x1e\xc2\x30", 16, 0) =
16
getrandom("\x48\x9e\xca\x06\x87\x23\xb3\x39\x33\x2c\xe9\x2d\x0e\x4f\xa0\x5a", 16, 0) =
16
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
read(0, "create 100 -1\n", 1024)        = 14
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f76e25bd8d0) = 8710
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
epoll_create1(EPOLL_CLOEXEC)            = 5
epoll_ctl(5, EPOLL_CTL_ADD, 4, {0, {u32=3898714816, u64=94733697395392}}) = 0
epoll_ctl(5, EPOLL_CTL_MOD, 4, {EPOLLIN, {u32=3898714816, u64=94733697395392}}) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f76e1da5000
mprotect(0x7f76e1da6000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f76e25a4d30, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|
CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tid=[8711], tls=0x7f76e25a5700, child_tidptr=0x7f76e25a59d0) = 8711
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
epoll_create1(EPOLL_CLOEXEC)            = 7
epoll_ctl(7, EPOLL_CTL_ADD, 6, {0, {u32=3898716944, u64=94733697397520}}) = 0
epoll_ctl(7, EPOLL_CTL_MOD, 6, {EPOLLIN, {u32=3898716944, u64=94733697397520}}) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f76e15a4000
mprotect(0x7f76e15a5000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f76e1da3d30, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|
CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tid=[8712], tls=0x7f76e1da4700, child_tidptr=0x7f76e1da49d0) = 8712
eventfd2(0, EFD_CLOEXEC)                = 8
fcntl(8, F_GETFL)                       = 0x2 (flags O_RDWR)
fcntl(8, F_SETFL, O_RDWR|O_NONBLOCK)    = 0
fcntl(8, F_GETFL)                       = 0x802 (flags O_RDWR|O_NONBLOCK)
fcntl(8, F_SETFL, O_RDWR|O_NONBLOCK)    = 0
poll([{fd=8, events=POLLIN}], 1, 0)     = 0 (Timeout)
socket(AF_NETLINK, SOCK_RAW|SOCK_CLOEXEC, NETLINK_ROUTE) = 9
bind(9, {sa_family=AF_NETLINK, nl_pid=0, nl_groups=00000000}, 12) = 0
getsockname(9, {sa_family=AF_NETLINK, nl_pid=8709, nl_groups=00000000}, [12]) = 0
```

```
sendto(9, {{len=20, type=RTM_GETLINK, flags=NLM_F_REQUEST|NLM_F_DUMP, seq=1633725592,
pid=0}, {ifi_family=AF_UNSPEC, ...}}, 20, 0, {sa_family=AF_NETLINK, nl_pid=0,
nl_groups=00000000}, 12) = 20
recvmsg(9, {msg_name={sa_family=AF_NETLINK, nl_pid=0, nl_groups=00000000},
msg_namelen=12, msg_iov=[{iov_base=[{{len=1316, type=RTM_NEWLINK, flags=NLM_F_MULTI,
seq=1633725592, pid=8709}, {ifi_family=AF_UNSPEC, ifi_type=ARPHRD_LOOPBACK,
ifi_index=if_nametoindex("lo"), ifi_flags=IFF_UP|IFF_LOOPBACK|IFF_RUNNING|
IFF_LOWER_UP, ifi_change=0}, [{{nla_len=7, nla_type=IFLA_IFNAME}, "lo"}, {{nla_len=8,
nla_type=IFLA_TXQLEN}, 1000}, {{nla_len=5, nla_type=IFLA_OPERSTATE}, 0}, {{nla_len=5,
nla_type=IFLA_LINKMODE}, 0}, {{nla_len=8, nla_type=IFLA_MTU}, 65536}, {{nla_len=8,
nla_type=IFLA_MIN_MTU}, 0}, {{nla_len=8, nla_type=IFLA_MAX_MTU}, 0}, {{nla_len=8,
nla_type=IFLA_GROUP}, 0}, {{nla_len=8, nla_type=IFLA_PROMISCUITY}, 0}, {{nla_len=8,
nla_type=IFLA_NUM_TX_QUEUES}, 1}, {{nla_len=8, nla_type=IFLA_GSO_MAX_SEGS}, 65535},
{{nla_len=8, nla_type=IFLA_GSO_MAX_SIZE}, 65536}, {{nla_len=8,
nla_type=IFLA_NUM_RX_QUEUES}, 1}, {{nla_len=5, nla_type=IFLA_CARRIER}, 1},
{{nla_len=12, nla_type=IFLA_QDISC}, "noqueue"}, {{nla_len=8,
nla_type=IFLA_CARRIER_CHANGES}, 0}, {{nla_len=5, nla_type=IFLA_PROTO_DOWN}, 0},
{{nla_len=8, nla_type=IFLA_CARRIER_UP_COUNT}, 0}, {{nla_len=8,
nla_type=IFLA_CARRIER_DOWN_COUNT}, 0}, {{nla_len=36, nla_type=IFLA_MAP}, {mem_start=0,
mem_end=0, base_addr=0, irq=0, dma=0, port=0}}, {{nla_len=10, nla_type=IFLA_ADDRESS},
"\x00\x00\x00\x00\x00\x00"}, {{nla_len=10, nla_type=IFLA_BROADCAST}, "\x00\x00\x00\
x00\x00\x00"}, {{nla_len=196, nla_type=IFLA_STATS64}, {rx_packets=876, tx_packets=876,
rx_bytes=77809, tx_bytes=77809, rx_errors=0, tx_errors=0, rx_dropped=0, tx_dropped=0,
multicast=0, collisions=0, rx_length_errors=0, rx_over_errors=0, rx_crc_errors=0,
rx_frame_errors=0, rx_fifo_errors=0, rx_missed_errors=0, tx_aborted_errors=0,
tx_carrier_errors=0, tx_fifo_errors=0, tx_heartbeat_errors=0, tx_window_errors=0,
rx_compressed=0, tx_compressed=0, rx_nohandler=0}}, {{nla_len=100,
nla_type=IFLA_STATS}, {rx_packets=876, tx_packets=876, rx_bytes=77809, tx_bytes=77809,
rx_errors=0, tx_errors=0, rx_dropped=0, tx_dropped=0, multicast=0, collisions=0,
rx_length_errors=0, rx_over_errors=0, rx_crc_errors=0, rx_frame_errors=0,
rx_fifo_errors=0, rx_missed_errors=0, tx_aborted_errors=0, tx_carrier_errors=0,
tx_fifo_errors=0, tx_heartbeat_errors=0, tx_window_errors=0, rx_compressed=0,
tx_compressed=0, rx_nohandler=0}}, {{nla_len=12, nla_type=IFLA_XDP}, {{nla_len=5,
nla_type=IFLA_XDP_ATTACHED}, XDP_ATTACHED_NONE}}, {{nla_len=760,
nla_type=IFLA_AF_SPEC}, [{{nla_len=136, nla_type=AF_INET}, {{nla_len=132,
nla_type=IFLA_INET_CONF}, [[IPV4_DEVCONF_FORWARDING-1] = 0,
[IPV4_DEVCONF_MC_FORWARDING-1] = 0, [IPV4_DEVCONF_PROXY_ARP-1] = 0,
[IPV4_DEVCONF_ACCEPT_REDIRECTS-1] = 1, [IPV4_DEVCONF_SECURE_REDIRECTS-1] = 1,
[IPV4_DEVCONF_SEND_REDIRECTS-1] = 1, [IPV4_DEVCONF_SHARED_MEDIA-1] = 1,
[IPV4_DEVCONF_RP_FILTER-1] = 0, [IPV4_DEVCONF_ACCEPT_SOURCE_ROUTE-1] = 1,
[IPV4_DEVCONF_BOOTP_RELAY-1] = 0, [IPV4_DEVCONF_LOG_MARTIANS-1] = 0,
[IPV4_DEVCONF_TAG-1] = 0, [IPV4_DEVCONF_ARPFILTER-1] = 0, [IPV4_DEVCONF_MEDIUM_ID-1] =
0, [IPV4_DEVCONF_NOXFRM-1] = 1, [IPV4_DEVCONF_NOPOLICY-1] = 1,
[IPV4_DEVCONF_FORCE_IGMP_VERSION-1] = 0, [IPV4_DEVCONF_ARP_ANNOUNCE-1] = 0,
[IPV4_DEVCONF_ARP_IGNORE-1] = 0, [IPV4_DEVCONF_PROMOTE_SECONDARIES-1] = 1,
[IPV4_DEVCONF_ARP_ACCEPT-1] = 0, [IPV4_DEVCONF_ARP_NOTIFY-1] = 0,
[IPV4_DEVCONF_ACCEPT_LOCAL-1] = 0, [IPV4_DEVCONF_SRC_VMARK-1] = 0,
[IPV4_DEVCONF_PROXY_ARP_PVLAN-1] = 0, [IPV4_DEVCONF_ROUTE_LOCALNET-1] = 0,
[IPV4_DEVCONF_IGMPV2_UNSOLICITED_REPORT_INTERVAL-1] = 10000,
[IPV4_DEVCONF_IGMPV3_UNSOLICITED_REPORT_INTERVAL-1] = 1000,
[IPV4_DEVCONF_IGNORE_ROUTES_WITH_LINKDOWN-1] = 0,
[IPV4_DEVCONF_DROP_UNICAST_IN_L2_MULTICAST-1] = 0, [IPV4_DEVCONF_DROP_GRATUITOUS_ARP-
1] = 0, [IPV4_DEVCONF_BC_FORWARDING-1] = 0]}}, {{nla_len=620, nla_type=AF_INET6},
[{{nla_len=8, nla_type=IFLA_INET6_FLAGS}, IF_READY}, {{nla_len=20,
nla_type=IFLA_INET6_CACHEINFO}, {max_reasm_len=65535, tstamp=280,
reachable_time=44684, retrans_time=1000}}, {{nla_len=208, nla_type=IFLA_INET6_CONF},
[[DEVCONF_FORWARDING] = 0, [DEVCONF_HOPLIMIT] = 64, [DEVCONF_MTU6] = 65536,
[DEVCONF_ACCEPT_RA] = 1, [DEVCONF_ACCEPT_REDIRECTS] = 1, [DEVCONF_AUTOCONF] = 1,
[DEVCONF_DAD_TRANSMITS] = 1, [DEVCONF_RTR_SOLICITS] = -1,
[DEVCONF_RTR_SOLICIT_INTERVAL] = 4000, [DEVCONF_RTR_SOLICIT_DELAY] = 1000,
[DEVCONF_USE_TEMPADDR] = -1, [DEVCONF_TEMP_VALID_LFT] = 604800,
[DEVCONF_TEMP_PREFERED_LFT] = 86400, [DEVCONF_REGEN_MAX_RETRY] = 3,
[DEVCONF_MAX_DESYNC_FACTOR] = 600, [DEVCONF_MAX_ADDRESSES] = 16,
[DEVCONF_FORCE_MLD_VERSION] = 0, [DEVCONF_ACCEPT_RA_DEFRTR] = 1,
[DEVCONF_ACCEPT_RA_PINFO] = 1, [DEVCONF_ACCEPT_RA_RTR_PREF] = 1,
[DEVCONF_RTR_PROBE_INTERVAL] = 60000, [DEVCONF_ACCEPT_RA_RT_INFO_MAX_PLEN] = 0,
[DEVCONF_PROXY_NDP] = 0, [DEVCONF_OPTIMISTIC_DAD] = 0, [DEVCONF_ACCEPT_SOURCE_ROUTE] =
0, [DEVCONF_MC_FORWARDING] = 0, [DEVCONF_DISABLE_IPV6] = 0, [DEVCONF_ACCEPT_DAD] = -1,
[DEVCONF_FORCE_TLLAO] = 0, [DEVCONF_NDISC_NOTIFY] = 0,
[DEVCONF_MLDV1_UNSOLICITED_REPORT_INTERVAL] = 10000,
[DEVCONF_MLDV2_UNSOLICITED_REPORT_INTERVAL] = 1000, ...]}, {{nla_len=300,
nla_type=IFLA_INET6_STATS}, [[IPSTATS_MIB_NUM] = 37, [IPSTATS_MIB_INPKTS] = 6,
[IPSTATS_MIB_INOCTETS] = 432, [IPSTATS_MIB_INDELIVERS] = 6,
[IPSTATS_MIB_OUTFORWDATAGRAMS] = 0, [IPSTATS_MIB_OUTPKTS] = 6, [IPSTATS_MIB_OUTOCTETS]
= 432, [IPSTATS_MIB_INHDRERRORS] = 0, [IPSTATS_MIB_INTOOBIGERRORS] = 0,
[IPSTATS_MIB_INNOROUTES] = 0, [IPSTATS_MIB_INADDRERRORS] = 0,
[IPSTATS_MIB_INUNKNOWNPROTOS] = 0, [IPSTATS_MIB_INTRUNCATEDPKTS] = 0,
[IPSTATS_MIB_INDISCARDS] = 0, [IPSTATS_MIB_OUTDISCARDS] = 0, [IPSTATS_MIB_OUTNOROUTES]
= 0, [IPSTATS_MIB_REASMTIMEOUT] = 0, [IPSTATS_MIB_REASMREQDS] = 0,
[IPSTATS_MIB_REASMOKS] = 0, [IPSTATS_MIB_REASMFAILS] = 0, [IPSTATS_MIB_FRAGOKS] = 0,
[IPSTATS_MIB_FRAGFAILS] = 0, [IPSTATS_MIB_FRAGCREATES] = 0, [IPSTATS_MIB_INMCASTPKTS]
= 0, [IPSTATS_MIB_OUTMCASTPKTS] = 2, [IPSTATS_MIB_INBCASTPKTS] = 0,
[IPSTATS_MIB_OUTBCASTPKTS] = 0, [IPSTATS_MIB_INMCASTOCTETS] = 0,
[IPSTATS_MIB_OUTMCASTOCTETS] = 152, [IPSTATS_MIB_INBCASTOCTETS] = 0,
[IPSTATS_MIB_OUTBCASTOCTETS] = 0, [IPSTATS_MIB_CSUMERRORS] = 0, ...]}, {{nla_len=52,
nla_type=IFLA_INET6_ICMP6STATS}, [[ICMP6_MIB_NUM] = 6, [ICMP6_MIB_INMSGS] = 2,
[ICMP6_MIB_INERRORS] = 0, [ICMP6_MIB_OUTMSGS] = 2, [ICMP6_MIB_OUTERRORS] = 0,
[ICMP6_MIB_CSUMERRORS] = 0]}, {{nla_len=20, nla_type=IFLA_INET6_TOKEN},
inet_pton(AF_INET6, "::")}, {{nla_len=5, nla_type=IFLA_INET6_ADDR_GEN_MODE},
IN6_ADDR_GEN_MODE_EUI64}]}]}]}, {{len=1320, type=RTM_NEWLINK, flags=NLM_F_MULTI,
seq=1633725592, pid=8709}, {ifi_family=AF_UNSPEC, ifi_type=ARPHRD_ETHER,
ifi_index=if_nametoindex("wlo1"), ifi_flags=IFF_UP|IFF_BROADCAST|IFF_RUNNING|
IFF_MULTICAST|IFF_LOWER_UP, ifi_change=0}, [{{nla_len=9, nla_type=IFLA_IFNAME},
"wlo1"}, {{nla_len=8, nla_type=IFLA_TXQLEN}, 1000}, {{nla_len=5,
nla_type=IFLA_OPERSTATE}, 6}, {{nla_len=5, nla_type=IFLA_LINKMODE}, 1}, {{nla_len=8,
nla_type=IFLA_MTU}, 1500}, {{nla_len=8, nla_type=IFLA_MIN_MTU}, 256}, {{nla_len=8,
nla_type=IFLA_MAX_MTU}, 2304}, {{nla_len=8, nla_type=IFLA_GROUP}, 0}, {{nla_len=8,
nla_type=IFLA_PROMISCUITY}, 0}, {{nla_len=8, nla_type=IFLA_NUM_TX_QUEUES}, 1},
{{nla_len=8, nla_type=IFLA_GSO_MAX_SEGS}, 65535}, {{nla_len=8,
nla_type=IFLA_GSO_MAX_SIZE}, 65536}, {{nla_len=8, nla_type=IFLA_NUM_RX_QUEUES}, 1},
{{nla_len=5, nla_type=IFLA_CARRIER}, 1}, {{nla_len=12, nla_type=IFLA_QDISC},
"noqueue"}, {{nla_len=8, nla_type=IFLA_CARRIER_CHANGES}, 8}, {{nla_len=5,
nla_type=IFLA_PROTO_DOWN}, 0}, {{nla_len=8, nla_type=IFLA_CARRIER_UP_COUNT}, 4},
{{nla_len=8, nla_type=IFLA_CARRIER_DOWN_COUNT}, 4}, {{nla_len=36, nla_type=IFLA_MAP},
{mem_start=0, mem_end=0, base_addr=0, irq=0, dma=0, port=0}}, {{nla_len=10,
nla_type=IFLA_ADDRESS}, "\x38\x00\x25\x36\x5b\x73"}, {{nla_len=10,
nla_type=IFLA_BROADCAST}, "\xff\xff\xff\xff\xff\xff"}, {{nla_len=196,
nla_type=IFLA_STATS64}, {rx_packets=50522, tx_packets=6842, rx_bytes=24212568,
tx_bytes=1022785, rx_errors=0, tx_errors=0, rx_dropped=0, tx_dropped=0, multicast=0,
collisions=0, rx_length_errors=0, rx_over_errors=0, rx_crc_errors=0,
rx_frame_errors=0, rx_fifo_errors=0, rx_missed_errors=0, tx_aborted_errors=0,
tx_carrier_errors=0, tx_fifo_errors=0, tx_heartbeat_errors=0, tx_window_errors=0,
rx_compressed=0, tx_compressed=0, rx_nohandler=0}}, {{nla_len=100,
nla_type=IFLA_STATS}, {rx_packets=50522, tx_packets=6842, rx_bytes=24212568,
tx_bytes=1022785, rx_errors=0, tx_errors=0, rx_dropped=0, tx_dropped=0, multicast=0,
collisions=0, rx_length_errors=0, rx_over_errors=0, rx_crc_errors=0,
rx_frame_errors=0, rx_fifo_errors=0, rx_missed_errors=0, tx_aborted_errors=0,
tx_carrier_errors=0, tx_fifo_errors=0, tx_heartbeat_errors=0, tx_window_errors=0,
rx_compressed=0, tx_compressed=0, rx_nohandler=0}}, {{nla_len=12, nla_type=IFLA_XDP},
{{nla_len=5, nla_type=IFLA_XDP_ATTACHED}, XDP_ATTACHED_NONE}}, {{nla_len=760,
nla_type=IFLA_AF_SPEC}, [{{nla_len=136, nla_type=AF_INET}, {{nla_len=132,
nla_type=IFLA_INET_CONF}, [[IPV4_DEVCONF_FORWARDING-1] = 0,
[IPV4_DEVCONF_MC_FORWARDING-1] = 0, [IPV4_DEVCONF_PROXY_ARP-1] = 0,
[IPV4_DEVCONF_ACCEPT_REDIRECTS-1] = 1, [IPV4_DEVCONF_SECURE_REDIRECTS-1] = 1,
[IPV4_DEVCONF_SEND_REDIRECTS-1] = 1, [IPV4_DEVCONF_SHARED_MEDIA-1] = 1,
[IPV4_DEVCONF_RP_FILTER-1] = 2, [IPV4_DEVCONF_ACCEPT_SOURCE_ROUTE-1] = 1,
[IPV4_DEVCONF_BOOTP_RELAY-1] = 0, [IPV4_DEVCONF_LOG_MARTIANS-1] = 0,
[IPV4_DEVCONF_TAG-1] = 0, [IPV4_DEVCONF_ARPFILTER-1] = 0, [IPV4_DEVCONF_MEDIUM_ID-1] =
0, [IPV4_DEVCONF_NOXFRM-1] = 0, [IPV4_DEVCONF_NOPOLICY-1] = 0,
[IPV4_DEVCONF_FORCE_IGMP_VERSION-1] = 0, [IPV4_DEVCONF_ARP_ANNOUNCE-1] = 0,
[IPV4_DEVCONF_ARP_IGNORE-1] = 0, [IPV4_DEVCONF_PROMOTE_SECONDARIES-1] = 1,
[IPV4_DEVCONF_ARP_ACCEPT-1] = 0, [IPV4_DEVCONF_ARP_NOTIFY-1] = 0,
[IPV4_DEVCONF_ACCEPT_LOCAL-1] = 0, [IPV4_DEVCONF_SRC_VMARK-1] = 0,
[IPV4_DEVCONF_PROXY_ARP_PVLAN-1] = 0, [IPV4_DEVCONF_ROUTE_LOCALNET-1] = 0,
[IPV4_DEVCONF_IGMPV2_UNSOLICITED_REPORT_INTERVAL-1] = 10000,
[IPV4_DEVCONF_IGMPV3_UNSOLICITED_REPORT_INTERVAL-1]
 = 1000, [IPV4_DEVCONF_IGNORE_ROUTES_WITH_LINKDOWN-1] = 0,
[IPV4_DEVCONF_DROP_UNICAST_IN_L2_MULTICAST-1] = 0, [IPV4_DEVCONF_DROP_GRATUITOUS_ARP-
1] = 0, [IPV4_DEVCONF_BC_FORWARDING-1] = 0]}}, {{nla_len=620, nla_type=AF_INET6},
[{{nla_len=8, nla_type=IFLA_INET6_FLAGS}, IF_READY}, {{nla_len=20,
nla_type=IFLA_INET6_CACHEINFO}, {max_reasm_len=65535, tstamp=806301,
reachable_time=16128, retrans_time=1000}}, {{nla_len=208, nla_type=IFLA_INET6_CONF},
[[DEVCONF_FORWARDING] = 0, [DEVCONF_HOPLIMIT] = 64, [DEVCONF_MTU6] = 1500,
[DEVCONF_ACCEPT_RA] = 0, [DEVCONF_ACCEPT_REDIRECTS] = 1, [DEVCONF_AUTOCONF] = 1,
[DEVCONF_DAD_TRANSMITS] = 1, [DEVCONF_RTR_SOLICITS] = -1,
[DEVCONF_RTR_SOLICIT_INTERVAL] = 4000, [DEVCONF_RTR_SOLICIT_DELAY] = 1000,
[DEVCONF_USE_TEMPADDR] = 2, [DEVCONF_TEMP_VALID_LFT] = 604800,
[DEVCONF_TEMP_PREFERED_LFT] = 86400, [DEVCONF_REGEN_MAX_RETRY] = 3,
[DEVCONF_MAX_DESYNC_FACTOR] = 600, [DEVCONF_MAX_ADDRESSES] = 16,
[DEVCONF_FORCE_MLD_VERSION] = 0, [DEVCONF_ACCEPT_RA_DEFRTR] = 1,
[DEVCONF_ACCEPT_RA_PINFO] = 1, [DEVCONF_ACCEPT_RA_RTR_PREF] = 1,
[DEVCONF_RTR_PROBE_INTERVAL] = 60000, [DEVCONF_ACCEPT_RA_RT_INFO_MAX_PLEN] = 0,
[DEVCONF_PROXY_NDP] = 0, [DEVCONF_OPTIMISTIC_DAD] = 0, [DEVCONF_ACCEPT_SOURCE_ROUTE] =
0, [DEVCONF_MC_FORWARDING] = 0, [DEVCONF_DISABLE_IPV6] = 0, [DEVCONF_ACCEPT_DAD] = 1,
[DEVCONF_FORCE_TLLAO] = 0, [DEVCONF_NDISC_NOTIFY] = 0,
[DEVCONF_MLDV1_UNSOLICITED_REPORT_INTERVAL] = 10000,
[DEVCONF_MLDV2_UNSOLICITED_REPORT_INTERVAL] = 1000, ...]}, {{nla_len=300,
nla_type=IFLA_INET6_STATS}, [[IPSTATS_MIB_NUM] = 37, [IPSTATS_MIB_INPKTS] = 3948,
[IPSTATS_MIB_INOCTETS] = 386720, [IPSTATS_MIB_INDELIVERS] = 3721,
[IPSTATS_MIB_OUTFORWDATAGRAMS] = 0, [IPSTATS_MIB_OUTPKTS] = 91,
[IPSTATS_MIB_OUTOCTETS] = 13338, [IPSTATS_MIB_INHDRERRORS] = 0,
[IPSTATS_MIB_INTOOBIGERRORS] = 0, [IPSTATS_MIB_INNOROUTES] = 0,
[IPSTATS_MIB_INADDRERRORS] = 0, [IPSTATS_MIB_INUNKNOWNPROTOS] = 0,
[IPSTATS_MIB_INTRUNCATEDPKTS] = 0, [IPSTATS_MIB_INDISCARDS] = 0,
[IPSTATS_MIB_OUTDISCARDS] = 3, [IPSTATS_MIB_OUTNOROUTES] = 0,
[IPSTATS_MIB_REASMTIMEOUT] = 0, [IPSTATS_MIB_REASMREQDS] = 0, [IPSTATS_MIB_REASMOKS] =
0, [IPSTATS_MIB_REASMFAILS] = 0, [IPSTATS_MIB_FRAGOKS] = 0, [IPSTATS_MIB_FRAGFAILS] =
0, [IPSTATS_MIB_FRAGCREATES] = 0, [IPSTATS_MIB_INMCASTPKTS] = 3948,
[IPSTATS_MIB_OUTMCASTPKTS] = 91, [IPSTATS_MIB_INBCASTPKTS] = 0,
[IPSTATS_MIB_OUTBCASTPKTS] = 0, [IPSTATS_MIB_INMCASTOCTETS] = 386720,
[IPSTATS_MIB_OUTMCASTOCTETS] = 13338, [IPSTATS_MIB_INBCASTOCTETS] = 0,
[IPSTATS_MIB_OUTBCASTOCTETS] = 0, [IPSTATS_MIB_CSUMERRORS] = 0, ...]}, {{nla_len=52,
nla_type=IFLA_INET6_ICMP6STATS}, [[ICMP6_MIB_NUM] = 6, [ICMP6_MIB_INMSGS] = 0,
[ICMP6_MIB_INERRORS] = 0, [ICMP6_MIB_OUTMSGS] = 40, [ICMP6_MIB_OUTERRORS] = 0,
[ICMP6_MIB_CSUMERRORS] = 0]}, {{nla_len=20, nla_type=IFLA_INET6_TOKEN},
inet_pton(AF_INET6, "::")}, {{nla_len=5, nla_type=IFLA_INET6_ADDR_GEN_MODE},
IN6_ADDR_GEN_MODE_NONE}]}]}]}], iov_len=4096}], msg_iovlen=1, msg_controllen=0,
msg_flags=0}, 0) = 2636
recvmsg(9, {msg_name={sa_family=AF_NETLINK, nl_pid=0, nl_groups=00000000},
msg_namelen=12, msg_iov=[{iov_base={{len=20, type=NLMSG_DONE, flags=NLM_F_MULTI,
seq=1633725592, pid=8709}, 0}, iov_len=4096}], msg_iovlen=1, msg_controllen=0,
msg_flags=0}, 0) = 20
sendto(9, {{len=20, type=RTM_GETADDR, flags=NLM_F_REQUEST|NLM_F_DUMP, seq=1633725593,
pid=0}, {ifa_family=AF_UNSPEC, ...}}, 20, 0, {sa_family=AF_NETLINK, nl_pid=0,
nl_groups=00000000}, 12) = 20
recvmsg(9, {msg_name={sa_family=AF_NETLINK, nl_pid=0, nl_groups=00000000},
msg_namelen=12, msg_iov=[{iov_base=[{{len=76, type=RTM_NEWADDR, flags=NLM_F_MULTI,
seq=1633725593, pid=8709}, {ifa_family=AF_INET, ifa_prefixlen=8,
ifa_flags=IFA_F_PERMANENT, ifa_scope=RT_SCOPE_HOST, ifa_index=if_nametoindex("lo")},
[{{nla_len=8, nla_type=IFA_ADDRESS}, inet_addr("127.0.0.1")}, {{nla_len=8,
nla_type=IFA_LOCAL}, inet_addr("127.0.0.1")}, {{nla_len=7, nla_type=IFA_LABEL}, "lo"},
{{nla_len=8, nla_type=IFA_FLAGS}, IFA_F_PERMANENT}, {{nla_len=20,
nla_type=IFA_CACHEINFO}, {ifa_prefered=4294967295, ifa_valid=4294967295, cstamp=280,
tstamp=280}}]}, {{len=88, type=RTM_NEWADDR, flags=NLM_F_MULTI, seq=1633725593,
pid=8709}, {ifa_family=AF_INET, ifa_prefixlen=24, ifa_flags=0,
ifa_scope=RT_SCOPE_UNIVERSE, ifa_index=if_nametoindex("wlo1")}, [{{nla_len=8,
nla_type=IFA_ADDRESS}, inet_addr("192.168.1.36")}, {{nla_len=8, nla_type=IFA_LOCAL},
inet_addr("192.168.1.36")}, {{nla_len=8, nla_type=IFA_BROADCAST},
inet_addr("192.168.1.255")}, {{nla_len=9, nla_type=IFA_LABEL}, "wlo1"}, {{nla_len=8,
nla_type=IFA_FLAGS}, IFA_F_NOPREFIXROUTE}, {{nla_len=20, nla_type=IFA_CACHEINFO},
{ifa_prefered=16123, ifa_valid=16123, cstamp=806332, tstamp=1684533}}]}],
iov_len=4096}], msg_iovlen=1, msg_controllen=0, msg_flags=0}, 0) = 164
recvmsg(9, {msg_name={sa_family=AF_NETLINK, nl_pid=0, nl_groups=00000000},
msg_namelen=12, msg_iov=[{iov_base=[{{len=72, type=RTM_NEWADDR, flags=NLM_F_MULTI,
seq=1633725593, pid=8709}, {ifa_family=AF_INET6, ifa_prefixlen=128,
ifa_flags=IFA_F_PERMANENT, ifa_scope=RT_SCOPE_HOST, ifa_index=if_nametoindex("lo")},
[{{nla_len=20, nla_type=IFA_ADDRESS}, inet_pton(AF_INET6, "::1")}, {{nla_len=20,
nla_type=IFA_CACHEINFO}, {ifa_prefered=4294967295, ifa_valid=4294967295, cstamp=280,
tstamp=280}}, {{nla_len=8, nla_type=IFA_FLAGS}, IFA_F_PERMANENT}]}, {{len=72,
type=RTM_NEWADDR, flags=NLM_F_MULTI, seq=1633725593, pid=8709}, {ifa_family=AF_INET6,
ifa_prefixlen=64, ifa_flags=IFA_F_PERMANENT, ifa_scope=RT_SCOPE_LINK,
ifa_index=if_nametoindex("wlo1")}, [{{nla_len=20, nla_type=IFA_ADDRESS},
inet_pton(AF_INET6, "fe80::e86a:cabf:fdaa:abe7")}, {{nla_len=20,
nla_type=IFA_CACHEINFO}, {ifa_prefered=4294967295, ifa_valid=4294967295,
cstamp=806301, tstamp=806470}}, {{nla_len=8, nla_type=IFA_FLAGS}, IFA_F_PERMANENT|
IFA_F_NOPREFIXROUTE}]}], iov_len=4096}], msg_iovlen=1, msg_controllen=0, msg_flags=0},
0) = 144
recvmsg(9, {msg_name={sa_family=AF_NETLINK, nl_pid=0, nl_groups=00000000},
msg_namelen=12, msg_iov=[{iov_base={{len=20, type=NLMSG_DONE, flags=NLM_F_MULTI,
seq=1633725593, pid=8709}, 0}, iov_len=4096}], msg_iovlen=1, msg_controllen=0,
msg_flags=0}, 0) = 20
close(9)                                = 0
socket(AF_INET, SOCK_STREAM|SOCK_CLOEXEC, IPPROTO_TCP) = 9
setsockopt(9, SOL_SOCKET, SO_REUSEADDR, [1], 4) = 0
bind(9, {sa_family=AF_INET, sin_port=htons(4140), sin_addr=inet_addr("127.0.0.1")},
16) = 0
listen(9, 100)                          = 0
getsockname(9, {sa_family=AF_INET, sin_port=htons(4140),
sin_addr=inet_addr("127.0.0.1")}, [128->16]) = 0
getsockname(9, {sa_family=AF_INET, sin_port=htons(4140),
sin_addr=inet_addr("127.0.0.1")}, [128->16]) = 0
write(6, "\1\0\0\0\0\0\0\0", 8)         = 8
write(8, "\1\0\0\0\0\0\0\0", 8)         = 8
poll([{fd=8, events=POLLIN}], 1, 0)     = 1 ([{fd=8, revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=8, events=POLLIN}], 1, 0)     = 0 (Timeout)
poll([{fd=8, events=POLLIN}], 1, 5000)  = 1 ([{fd=8, revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=8, events=POLLIN}], 1, 0)     = 0 (Timeout)
write(6, "\1\0\0\0\0\0\0\0", 8)         = 8
poll([{fd=8, events=POLLIN}], 1, -1)    = 1 ([{fd=8, revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=8, events=POLLIN}], 1, 0)     = 1 ([{fd=8, revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=8, events=POLLIN}], 1, 0)     = 0 (Timeout)
write(6, "\1\0\0\0\0\0\0\0", 8)         = 8
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "OK: 8710\n", 9)               = 9
read(0, "exit\n", 1024)                 = 5
poll([{fd=8, events=POLLIN}], 1, 0)     = 0 (Timeout)
write(6, "\1\0\0\0\0\0\0\0", 8)         = 8
poll([{fd=8, events=POLLIN}], 1, -1)    = 1 ([{fd=8, revents=POLLIN}])
read(8, "\1\0\0\0\0\0\0\0", 8)          = 8
poll([{fd=8, events=POLLIN}], 1, 0)     = 0 (Timeout)
poll([{fd=8, events=POLLIN}], 1, 0)     = 0 (Timeout)
write(6, "\1\0\0\0\0\0\0\0", 8)         = 8
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=8710, si_uid=1000,
si_status=0, si_utime=0, si_stime=0} ---
lseek(0, -1, SEEK_CUR)                  = -1 ESPIPE (Illegal seek)
exit_group(0)                           = ?
+++ exited with 0 +++
```

6. Выводы

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

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

Полученные мной навыки работы с очередями сообщений можно использовать при проектировании различных мультипроцессорных программ.