

Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

Лабораторные работы №5-7 по курсу

«Операционные системы»

Группа: М80-206Б-20

Студент: Шипилова Т.П.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 29.12.23

Москва, 2023

Постановка задачи

Вариант 2.

Цель работы

Целью является приобретение практических навыков в:

1. Управлении серверами сообщений (№5)
2. Применение отложенных вычислений (№6)
3. Интеграция программных систем друг с другом (№7)

Задание

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 той топологией, которая определена вариантом. Связь между узлами необходимо осуществить при помощи технологии очередей сообщений. Также в данной системе необходимо предусмотреть проверку доступности узлов в соответствии с вариантом. При убийстве («kill -9») любого вычислительного узла система должна пытаться максимально сохранять свою работоспособность, а именно все дочерние узлы убитого узла могут стать недоступными, но родительские узлы должны сохранить свою работоспособность.

Управляющий узел отвечает за ввод команд от пользователя и отправку этих команд на вычислительные узлы. Список основных поддерживаемых команд:

Создание нового вычислительного узла

Формат команды: `create id [parent]`

`id` – целочисленный идентификатор нового вычислительного узла

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

Формат вывода:

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

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

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

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

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

Пример:

```
> create 10 5
```

Ok: 3128

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

Исполнение команды на вычислительном узле

Формат команды: `exec id [params]`

id – целочисленный идентификатор вычислительного узла, на который отправляется команда

Формат вывода:

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

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

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

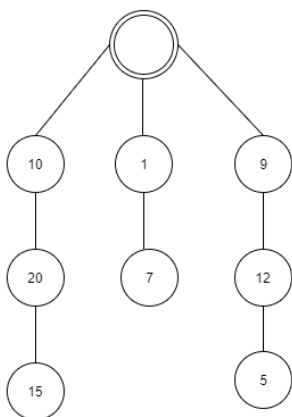
«Error:id: [Custom error]» - любая другая обрабатываемая ошибка

Пример:

Можно найти в описании конкретной команды, определенной вариантом задания.

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

Топология 2, узлы находятся в дереве общего вида



Набора команд 4 (поиск подстроки в строке)

Формат команды:

> `exec id`

> `text_string`

> `pattern_string`

[result] – номера позиций, где найден образец, разделенный точкой с запятой

text_string — текст, в котором искать образец. Алфавит: [A-Za-z0-9]. Максимальная длина строки 108 символов

pattern_string — образец

Пример:

```
> exes 10
```

```
> abracadabra
```

```
> abra
```

```
Ok:10:0;7
```

```
> exes 10
```

```
> abracadabra
```

```
> mmm
```

```
Ok:10: -1
```

Тип проверки доступности узлов - НЕ РЕАЛИЗОВАНО!

Формат команды: ping id

Команда проверяет доступность конкретного узла. Если узла нет, то необходимо выводить ошибку: «Error: Not found»

Пример:

```
> ping 10
```

```
Ok: 1 // узел 10 доступен
```

```
> ping 17
```

```
Ok: 0 // узел 17 недос
```

Общий метод и алгоритм решения

Имеем два исполняемых файла, но непосредственно работаем только с одним. Другой используется внутри вызова `exec1` для создания дочернего вычислительного узла. Нам нужно организовать асинхронность, для этого запускаем в клиенте несколько потоков конвейерообразно. Первый работает с пользователем, принимает от него команды и через `push-pull` сокет передает в поток, отвечающий за отправку сообщения и получение результата. И если данный поток может надолго заснуть в ожидании ответа от вычислительного узла, первый потенциально никогда не заблокируется надолго и сможет продолжать работать с пользователем, заполняя очередь для следующего потока. Кроме этого есть поток, отвечающий за вывод результатов запросов и поток, отвечающий за регистрацию свежесозданных узлов. В дереве общего вида мы не знаем, где расположена узел, на который идет запрос и как его найти, в отличие от бинарного. Так что каждый узел, начиная с управляющего, при необходимости отправить сообщение проходит по всем дочерним узлам, подключается и отправляет сообщение. Далее узлы выполняют запрос, если он адресован им, или пробрасывают его дальше по дереву. Так запрос волной пройдет по всем узлам и точно найдет тот, которому он адресован, если он, конечно, доступен.

1. `main.cpp` - файл, в котором происходит запуск приложения, с помощью вызова метода `run` у класса `SpringBootApplication`
2. `server.cpp` - код серверов(вычислительных узлов).
3. `MessageData.hpp` - класс, определяющий содержание сообщения.
4. `ZmqUtils.hpp` - класс со статическими утилитными методами.
5. `MessageTypes.hpp` - enum, содержащий типы сообщений.
6. `MessageBuilder.hpp` - класс, собирающий сообщения по требованию. В нем также содержатся средства сериализации-десериализации.
7. `Message.hpp` - класс сообщения.
8. `AbstractNode.hpp` - абстрактный узел.
9. `SpringBootApplication.hpp` - клиент(управляющий узел). Наследуется от абстрактного.
10. `ServerNode.hpp` - сервер(вычислительный узел). Наследуется от абстрактного.
11. `ChildNodeInfo.hpp` - класс, содержащий представление дочернего узла при хранении в мане потомков данного.

Использованные системные вызовы

`exec1()` - для запуска вложенного кода.

Код программы

Код программы находится в репозитории на githab:

<https://github.com/TPShipilova/OS3sem>

Протокол работы программы

Тестирование:

```
./lab
create 1 -1
add new node 1 to parent -1
received Pid: 24696
awaited
server created
SERVER CREATED
from registrator 24696
Ok:24696
exec 1 mammamia
ma
1 mammamia ma
0 3
RECEIVED!! mammamia ma
Exec Ok: 1 0 3
create 1 3
add new node 1 to parent 3
already exist
another message
create 3 1
add new node 3 to parent 1
register message got
server created
SERVER CREATED
from registrator 25424
Ok:25424
create 2 3
add new node 2 to parent 3
register message got
server created
SERVER CREATED
from registrator 25554
Ok:25554
```

Strace:

```
strace ./server
execve("./server", ["/server"], 0x7ffe235271f0 /* 74 vars */) = 0
brk(NULL)                                = 0x55c2d368a000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc5992470) = -1 EINVAL (Недопустимый аргумент)
```

```

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7f5c81467000
access("/etc/ld.so.preload", R_OK)          = -1 ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=71451, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 71451, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f5c81455000
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\3\0>\0\1\0\0\0\240\233\1\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=634936, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 636784, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f5c813b9000
mmap(0x7f5c813d1000, 397312, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000) = 0x7f5c813d1000
mmap(0x7f5c81432000, 106496, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x79000) = 0x7f5c81432000
mmap(0x7f5c8144c000, 36864, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x92000) = 0x7f5c8144c000
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\3\0>\0\1\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) =
0x7f5c81000000
mprotect(0x7f5c8109a000, 1576960, PROT_NONE) = 0
mmap(0x7f5c8109a000, 1118208, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9a000) = 0x7f5c8109a000
mmap(0x7f5c811ab000, 454656, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1ab000) = 0x7f5c811ab000
mmap(0x7f5c8121b000, 57344, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x21a000) = 0x7f5c8121b000
mmap(0x7f5c81229000, 10432, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f5c81229000
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\3\0>\0\1\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) =
0x7f5c81399000

mmap(0x7f5c8139c000, 94208, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f5c8139c000

mmap(0x7f5c813b3000, 16384, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000) = 0x7f5c813b3000

mmap(0x7f5c813b7000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d000) = 0x7f5c813b7000

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\3\0>\0\1\0\0\0P\237\2\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"..., 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"..., 48, 848) = 48

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0 =\340\256\3\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"..., 784, 64) = 784

mmap(NULL, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f5c80c00000

mmap(0x7f5c80c28000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f5c80c28000

mmap(0x7f5c80dbd000, 360448, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f5c80dbd000

mmap(0x7f5c80e15000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x214000) = 0x7f5c80e15000

mmap(0x7f5c80e1b000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f5c80e1b000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libbsd.so.0", O_RDONLY|O_CLOEXEC) = 3

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

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=89096, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 94432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f5c81381000

mprotect(0x7f5c81385000, 69632, PROT_NONE) = 0

mmap(0x7f5c81385000, 53248, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7f5c81385000

```

```

mmap(0x7f5c81392000, 12288, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x11000) = 0x7f5c81392000

mmap(0x7f5c81396000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x14000) = 0x7f5c81396000

mmap(0x7f5c81398000, 224, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f5c81398000

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\3\0>\0\1\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) =
0x7f5c81329000

mprotect(0x7f5c81335000, 303104, PROT_NONE) = 0

mmap(0x7f5c81335000, 229376, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xc000) = 0x7f5c81335000

mmap(0x7f5c8136d000, 69632, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x44000) = 0x7f5c8136d000

mmap(0x7f5c8137f000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x55000) = 0x7f5c8137f000

close(3) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7f5c81327000

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\3\0>\0\1\0\0\0\340L\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) =
0x7f5c812d6000

mmap(0x7f5c812da000, 172032, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7f5c812da000

mmap(0x7f5c81304000, 118784, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2e000) = 0x7f5c81304000

mmap(0x7f5c81321000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4a000) = 0x7f5c81321000

mmap(0x7f5c81323000, 14416, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f5c81323000

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\3\0>\0\1\0\0\0\255\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) =
0x7f5c80ed5000

mprotect(0x7f5c80edf000, 446464, PROT_NONE) = 0

mmap(0x7f5c80edf000, 286720, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xa000) = 0x7f5c80edf000

mmap(0x7f5c80f25000, 155648, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x50000) = 0x7f5c80f25000

mmap(0x7f5c80f4c000, 16384, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x76000) = 0x7f5c80f4c000

mmap(0x7f5c80f50000, 719360, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f5c80f50000

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\3\0>\0\1\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) =
0x7f5c81282000

mprotect(0x7f5c8128d000, 282624, PROT_NONE) = 0

mmap(0x7f5c8128d000, 229376, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xb000) = 0x7f5c8128d000

mmap(0x7f5c812c5000, 49152, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x43000) = 0x7f5c812c5000

mmap(0x7f5c812d2000, 16384, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4f000) = 0x7f5c812d2000

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\3\0>\0\1\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) =
0x7f5c80b19000

mmap(0x7f5c80b27000, 507904, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7f5c80b27000

mmap(0x7f5c80ba3000, 372736, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x7f5c80ba3000

mmap(0x7f5c80bfe000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x7f5c80bfe000

```

```

close(3)                = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libmd.so.0", O_RDONLY|O_CLOEXEC) = 3

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

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=47472, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 49384, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f5c81275000

mmap(0x7f5c81277000, 28672, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f5c81277000

mmap(0x7f5c8127e000, 8192, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9000) = 0x7f5c8127e000

mmap(0x7f5c81280000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xa000) = 0x7f5c81280000

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\3\0>\0\1\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) = 0x7f5c81273000

mmap(NULL, 16424, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f5c8126e000

mmap(0x7f5c8126f000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7f5c8126f000

mmap(0x7f5c81270000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f5c81270000

mmap(0x7f5c81271000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f5c81271000

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\3\0>\0\1\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) =
0x7f5c80a4e000

mprotect(0x7f5c80a6f000, 634880, PROT_NONE) = 0

mmap(0x7f5c80a6f000, 380928, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x21000) = 0x7f5c80a6f000

mmap(0x7f5c80acc000, 249856, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7e000) = 0x7f5c80acc000

```

```

mmap(0x7f5c80b0a000, 61440, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xbb000) = 0x7f5c80b0a000
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\3\0>\0\1\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) =
0x7f5c8123f000
mprotect(0x7f5c81243000, 163840, PROT_NONE) = 0
mmap(0x7f5c81243000, 110592, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7f5c81243000
mmap(0x7f5c8125e000, 49152, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1f000) = 0x7f5c8125e000
mmap(0x7f5c8126b000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2b000) = 0x7f5c8126b000
mmap(0x7f5c8126d000, 56, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f5c8126d000
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\3\0>\0\1\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) =
0x7f5c81239000
mmap(0x7f5c8123b000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f5c8123b000
mmap(0x7f5c8123c000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f5c8123c000
mmap(0x7f5c8123d000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f5c8123d000
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\3\0>\0\1\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) =
0x7f5c80ec7000
mprotect(0x7f5c80eca000, 36864, PROT_NONE) = 0

```

```

mmap(0x7f5c80eca000, 24576, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f5c80eca000

mmap(0x7f5c80ed0000, 8192, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9000) = 0x7f5c80ed0000

mmap(0x7f5c80ed3000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xb000) = 0x7f5c80ed3000

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\3\0>\0\1\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) = 0x7f5c81237000

mmap(NULL, 24592, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f5c81230000

mmap(0x7f5c81232000, 8192, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f5c81232000

mmap(0x7f5c81234000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7f5c81234000

mmap(0x7f5c81235000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7f5c81235000

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\3\0>\0\1\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) =
0x7f5c80eb3000

mmap(0x7f5c80eb6000, 40960, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f5c80eb6000

mmap(0x7f5c80ec0000, 12288, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xd000) = 0x7f5c80ec0000

mmap(0x7f5c80ec3000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xf000) = 0x7f5c80ec3000

mmap(0x7f5c80ec5000, 6728, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f5c80ec5000

close(3) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7f5c8122e000

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7f5c80eb0000

```

```

arch_prctl(ARCH_SET_FS, 0x7f5c80eb09c0) = 0
set_tid_address(0x7f5c80eb0c90)          = 29025
set_robust_list(0x7f5c80eb0ca0, 24)      = 0
rseq(0x7f5c80eb1360, 0x20, 0, 0x53053053) = 0
mprotect(0x7f5c80e15000, 16384, PROT_READ) = 0
mprotect(0x7f5c80ec3000, 4096, PROT_READ) = 0
mprotect(0x7f5c81235000, 4096, PROT_READ) = 0
mprotect(0x7f5c80ed3000, 4096, PROT_READ) = 0
mprotect(0x7f5c8123d000, 4096, PROT_READ) = 0
mprotect(0x7f5c8126b000, 4096, PROT_READ) = 0
mprotect(0x7f5c80b0a000, 53248, PROT_READ) = 0
mprotect(0x7f5c81271000, 4096, PROT_READ) = 0
mprotect(0x7f5c81280000, 4096, PROT_READ) = 0
mprotect(0x7f5c80bfe000, 4096, PROT_READ) = 0
mprotect(0x7f5c812d2000, 8192, PROT_READ) = 0
mprotect(0x7f5c813b7000, 4096, PROT_READ) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7f5c8122c000
mprotect(0x7f5c8121b000, 45056, PROT_READ) = 0
mprotect(0x7f5c80f4c000, 12288, PROT_READ) = 0
mprotect(0x7f5c81321000, 4096, PROT_READ) = 0
mprotect(0x7f5c8137f000, 4096, PROT_READ) = 0
mprotect(0x7f5c81396000, 4096, PROT_READ) = 0
mprotect(0x7f5c8144c000, 32768, PROT_READ) = 0
mprotect(0x55c2d345f000, 4096, PROT_READ) = 0
mprotect(0x7f5c814a1000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY})
= 0
munmap(0x7f5c81455000, 71451)            = 0
getrandom("\xaf\x99\xe5\x48\x6d\xf2\x5d\x53", 8, GRND_NONBLOCK) = 8
brk(NULL)                                = 0x55c2d368a000
brk(0x55c2d36ab000)                      = 0x55c2d36ab000
futex(0x7f5c8122977c, FUTEX_WAKE_PRIVATE, 2147483647) = 0

```

```
rt_sigaction(SIGCHLD, {sa_handler=0x55c2d3451d54, sa_mask=[CHLD],
sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x7f5c80c42520},
{sa_handler=SIG_DFL, sa_mask=[], sa_flags=0}, 8) = 0
```

```
exit_group(0)                = ?
```

```
+++ exited with 0 +++
```


Вывод

В ходе лабораторной работы я познакомилась с тем, как можно построить распределенную асинхронную систему с помощью очередей сообщений. Более того, программа сделана таким образом, что при убийстве какого-нибудь вычислительного узла его потомки станут недоступны, но мы можем создавать новые узлы и работать с уже созданными дальше.