**Отчет**

Предмет - *операционные системы*

Лабораторная работа *6*

Выполнил — *студент группы М80-208Б-17 Захаров Игорь*

Тема: Работа с серверами сообщений

Цель:

Управлении серверами сообщений

Применение отложенных вычислений

Задание:

Реализовать клиент-серверную систему по асинхронной обработке запросов. Необходимо составить программы сервера и клиента. При запуске сервер и клиент должны быть настраиваемы, то есть должна быть возможность поднятия на одной ЭВМ нескольких серверов по обработке данных и нескольких клиентов, которые к ним относятся. Все общение между процессами сервера и клиентов должно осуществляться через сервер сообщений.

Серверное приложение – банк. Клиентское приложение клиент банка. Клиент может отправить какую-то денежную сумму в банк на хранение. Клиент также может запросить из банка произвольную сумму. Клиенты могут посылать суммы на счета других клиентов. Запросить собственный счет. При снятии должна производиться проверка на то, что у клиента достаточно денег для снятия денежных средств. Идентификатор клиента задается во время запуска клиентского приложения, как и адрес банка. Считать, что идентификаторы при запуске клиентов будут уникальными.

*Вариант задания*:

Сервер сообщений : ZeroMQ ;

Структура данных - бинарное дерево , тип ключа - 32 битное число

*Дополнительная возможность сервера*:

Сохранение данных о счетах клиентов при завершении работы сервера и возобновлении

Решение:

Бинарное дерево было реализовано следуюущим образом :

Написана функция вставки клиента в дерево , удаления из дерева и поиска по дереву , остальные функции : показать счет , внести деньги - построенны на этих функциях .

В программе клиента используются функции из библиотеки zmq.h : zmq\_connect - для подклчению к удаленному адресу , zmq\_ctx\_new - создание контекста , zmq\_socket - создание сокета в этом контексте . Далее программа клиента отправляет запросы с помощью zmq\_msg\_send , и принимает ответ с помощью zmq\_msg\_recv .

Программа сервера также создает контекст и сокет , но сокет присоединяет к локальному адресу с помощью zmq\_bind . Для сохранения данных о клиентах использовал файл : при запуске сервера происходит считывание данных с файла в массив и удаление файла , в конце измененный массив записывается в файл , в итоге все данные о клиентах сохраняются .

Сценарий выполнения работы:

server

go!

id : 112 ; insert sum : 100

go!

id : 112 ; insert sum : 20

go!

id : 111 ; insert sum : 200

go!

go!

id 112 sum 100

id 111 sum 200

client 111

Client Starting….

enter adress of server : 4040

enter:

1) to insert money

2) to get money

3) to transfer money to another client

4) to show cash

-1) to exit

1 200

enter sum of money: successful inserting money

-1

exit client

client 112

Client Starting….

enter adress of server : 4040

enter:

1) to insert money

2) to get money

3) to transfer money to another client

4) to show cash

-1) to exit

1 100

enter sum of money: successful inserting money

1 20

enter sum of money: successful inserting money

-1

exit client

server

go!

id : 111 ; insert sum : 10

go!

id : 111 ; sum on accaunt : 210

go!

id : 112 ; insert sum : 10

go!

id : 112 ; sum on accaunt : 110

go!

id : 112 to id : 111 ; transfer sum : 10

go!

id : 112 ; sum on accaunt : 100

go!

go!

id 112 sum 100

id 111 sum 220

client 112

Client Starting….

enter adress of server : 4040

enter:

1) to insert money

2) to get money

3) to transfer money to another client

4) to show cash

-1) to exit

1 10

enter sum of money: successful inserting money

4

your cash is : 210

-1

exit client

client 111

Client Starting….

enter adress of server : 4040

enter:

1) to insert money

2) to get money

3) to transfer money to another client

4) to show cash

-1) to exit

1 10

enter sum of money: successful inserting money

4

your cash is : 110

3

enter sum of money: 10

enter id of client: 111

successful transmiting

4

your cash is : 100

-1

exit client

Листинг программы :

CLIENT.C

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <zmq.h>

#include "message.h"

void usage\_user(){

printf("enter:\n");

printf("1) to insert money\n");

printf("2) to get money\n");

printf("3) to transfer money to another client\n");

printf("4) to show cash\n");

printf("-1) to exit\n");

}

void send\_recv(message \*mes){

zmq\_msg\_t request;

zmq\_msg\_init\_size(&request, sizeof(message));

memcpy(zmq\_msg\_data(&request), mes, sizeof(message));

zmq\_msg\_send(&request, mes->requester, 0);

zmq\_msg\_close(&request);

zmq\_msg\_init(&request);

zmq\_msg\_recv(&request, mes->requester, 0);

mes = (message\*)zmq\_msg\_data(&request);

printf("%s\n", mes->text);

zmq\_msg\_close(&request);

}

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

{

message mes;

char txt[30];

void\* context = zmq\_ctx\_new();

if(context == NULL)

return 0;

int adress;

void\* request = zmq\_socket(context, ZMQ\_REQ);

if(request == NULL)

return 0;

printf("Client Starting….\n");

printf("enter adress of server : ");

scanf("%d",&adress);

sprintf(txt,"%s%d","tcp://localhost:",adress);

if(zmq\_connect(request, txt)){

perror("zmq\_connect");

exit(0);

}//create socket here

usage\_user();

char act[20];

do{

scanf("%s",act);

mes.action=atoi(act);

switch(mes.action){

case 1 :

printf("enter sum of money: ");

scanf("%d",&mes.sum);

if(mes.sum<=0){

printf("enter sum > 0\n");

break;

}

mes.id=atoi(argv[1]);

mes.requester=request;

send\_recv(&mes);break;

case 2 :

printf("enter sum of money: ");

scanf("%d",&mes.sum);

if(mes.sum<=0){

printf("enter sum > 0\n");

break;

}

mes.id=atoi(argv[1]);

mes.requester=request;

send\_recv(&mes);break;

case 3 :

printf("enter sum of money: ");

scanf("%d",&mes.sum);

printf("enter id of client: ");

scanf("%d",&mes.receiverClient);

mes.id=atoi(argv[1]);

mes.requester=request;

send\_recv(&mes);break;

case 4 :

mes.id=atoi(argv[1]);

mes.requester=request;

send\_recv(&mes);break;

default:break;

}

}

while(mes.action!=-1);

mes.id=atoi(argv[1]);

send\_recv(&mes);

zmq\_close(request);

zmq\_ctx\_destroy(context);

return 0;

}

SERVER.C

#include <string.h>

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <signal.h>

#include <time.h>

#include <sys/ioctl.h>

#include <stropts.h>

#include "zmq.h"

#include "message.h"

#include "bst.h"

#define ppl\_number 100

#define file\_name "file.txt"

volatile sig\_atomic\_t flag = 0;

typedef struct ppls{

int id;

int money;

}PPL;

void quit\_server\_func()

{

printf("\nserver quit\n");

flag = 3;

}

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

{

int cash,num\_of\_people=0,index=0;

char txt[20];

int counter=0;

PPL ppl[ppl\_number];

FILE\* fd = fopen(file\_name,"rw+");

if(fd!=NULL){

while(fscanf(fd,"%d%d",&ppl[counter].id,&ppl[counter].money)!=EOF){

counter++;

}

}

remove(file\_name);

index=counter;

BST Persons=NULL;

zmq\_msg\_t request;

zmq\_msg\_t reply;

void\* context = zmq\_ctx\_new();

if(context == NULL)

return 0;

message \*mes;

void\* respond = zmq\_socket(context, ZMQ\_REP);

if(respond == NULL)

return 0;

if(argc==2){

sprintf(txt,"%s%s","tcp://\*:",argv[1]);

if(zmq\_bind(respond, txt)==-1)

return 0;

}

else

if(zmq\_bind(respond, "tcp://\*:4040")==-1)

return 0;

signal(SIGINT,quit\_server\_func);

while(flag!=3){

zmq\_msg\_init(&request);

zmq\_msg\_recv(&request, respond, 0);

mes=(message\*)zmq\_msg\_data(&request);

zmq\_msg\_close(&request);

zmq\_msg\_init\_size(&reply,sizeof(message));

if(flag==3)break;

int flag\_1=0,ind=0;

switch(mes->action){

case 1 :

if(!find(Persons,mes->id)){

for(counter=0;counter<index;counter++){

if(ppl[counter].id==mes->id){

ppl[counter].money+=mes->sum;

flag\_1=1;

ind=counter;

}

}

if(flag\_1==0){

ppl[index].id=mes->id;

ppl[index].money=mes->sum;

index++;

}

num\_of\_people++;

}

if(flag\_1>0)

add(&Persons,mes->id,ppl[ind].money);

else

add(&Persons,mes->id,mes->sum);

strcpy(mes->text,"successful inserting money");

printf("id : %d ; insert sum : %d\n",mes->id,mes->sum);

memcpy(zmq\_msg\_data(&reply),mes,sizeof(message));

zmq\_msg\_send(&reply, respond, 0);

zmq\_msg\_close(&reply); break;

case 2 :

if(get\_money(Persons,mes->id,mes->sum)){

counter=0;

for(counter;ppl[counter].id!=mes->id;counter++){}

ppl[counter].money-=mes->sum;

strcpy(mes->text,"successful getting money");

printf("id : %d ; get sum : %d\n",mes->id,mes->sum);

}

else{

printf("id : %d ; money < ask \n",mes->id);

strcpy(mes->text,"money on accaunt less than you ask");

}

memcpy(zmq\_msg\_data(&reply),mes,sizeof(message));

zmq\_msg\_send(&reply, respond, 0);

zmq\_msg\_close(&reply); break;

case 3 :

if(find(Persons,mes->receiverClient)){

if(get\_money(Persons,mes->id,mes->sum)){

counter=0;

for(counter;ppl[counter].id!=mes->id;counter++){}

ppl[counter].money-=mes->sum;

add(&Persons,mes->receiverClient,mes->sum);

counter=0;

for(counter;ppl[counter].id!=mes->receiverClient;counter++){}

ppl[counter].money+=mes->sum;

strcpy(mes->text,"successful transmiting");

printf("id : %d to id : %d ; transfer sum : %d\n",mes->id,mes->receiverClient,mes->sum);

}

}

else{

strcpy(mes->text,"no such client");

printf("id : %d ; dosnt exist\n",mes->receiverClient);

}

memcpy(zmq\_msg\_data(&reply),mes,sizeof(message));

zmq\_msg\_send(&reply, respond, 0);

zmq\_msg\_close(&reply); break;

case 4 :

cash=show\_cash(Persons,mes->id);

printf("id : %d ; sum on accaunt : %d\n",mes->id,cash);

sprintf(mes->text,"%s %d","your cash is :",cash);

memcpy(zmq\_msg\_data(&reply),mes,sizeof(message));

zmq\_msg\_send(&reply, respond, 0);

zmq\_msg\_close(&reply);break;

case -1 :

if(find(Persons,mes->id)){

num\_of\_people--;

}

del(Persons,mes->id);

strcpy(mes->text,"exit client");

memcpy(zmq\_msg\_data(&reply),mes,sizeof(message));

zmq\_msg\_send(&reply, respond, 0);

zmq\_msg\_close(&reply);

break;

}

if(!num\_of\_people)

flag=3;

}

counter=0;

fd = fopen(file\_name,"a+");

while(counter!=index){

fprintf(fd,"%d %d ",ppl[counter].id,ppl[counter].money);

counter++;

}

for(counter=0;counter<index;counter++){

printf("id %d sum %d\n",ppl[counter].id,ppl[counter].money);

}

zmq\_close(respond);

zmq\_ctx\_destroy(context);

return 0;

}

BST.C

#include "bst.h"

int show\_cash(BST root,int id){

BST tmp = find(root,id);

if(tmp)

return tmp->money;

else

return 0;

}

int get\_money(BST root,int id , int money){

BST tmp = find(root,id);

if(!tmp)

return 0;

if(tmp->money>=money){

tmp->money-=money;

}

else

return 0;

return 1;

}

void add(BST \*root, int newid , int money)

{

if (!(\*root)) {

BST newNode = (BST) malloc(sizeof(\*newNode));

if (!newNode) {

printf("Error: no memory\n");

exit(FAILURE);

}

newNode->left = newNode->right = NULL;

newNode->id = newid;

newNode->money = money;

\*root = newNode;

return;

}

if(newid == (\*root)->id){

(\*root)->money+=money;

} else if (newid < (\*root)->id){

add(&(\*root)->left, newid , money);

} else if(newid > (\*root)->id){

add(&(\*root)->right, newid , money);

}

}

BST find(BST root, int id)

{

if (!root) {

return root;

}

if (id < root->id) {

return find(root->left, id);

} else if (id > root->id) {

return find(root->right, id);

} else {

return root;

}

}

BST minValueNode(BST root)

{

BST cur = root;

while (cur->left)

cur = cur->left;

return cur;

}

BST del(BST root, int id)

{

if (!root)

return root;

if (id < root->id) {

root->left = del(root->left, id);

}

else if (id > root->id) {

root->right = del(root->right, id);

}

else {

if (!root->left) {

BST tmp = root->right;

free(root);

root = NULL;

return tmp;

}

else if (!root->right) {

BST tmp = root->left;

free(root);

root = NULL;

return tmp;

}

BST tmp2,tmp = minValueNode(root->right);

root->id = tmp->id;

root->money=tmp->money;

tmp=root->right;

while(tmp2!=tmp||tmp2->left!=tmp){

tmp2=tmp2->left;

}

if(tmp2==tmp){

root->right==tmp->right;

free(tmp);

}

else{

tmp2->left=tmp->right;

free(tmp);

}

}

return root;

}

bool empty(BST root)

{

return !root;

}

Отчет Strace

./client

MacBook-Air-MacBook-3:os6 macbookair$ sudo dtruss ./client 10

dtrace: system integrity protection is on, some features will not be available

SYSCALL(args) = return

Client Starting….

enter adress of server : open("/dev/dtracehelper\0", 0x2, 0xFFFFFFFFE00DF160) = 3 0

ioctl(0x3, 0x80086804, 0x7FFEE00DEF70) = 0 0

close(0x3) = 0 0

access("/AppleInternal/XBS/.isChrooted\0", 0x0, 0x0) = -1 Err#2

bsdthread\_register(0x7FFF5DC87408, 0x7FFF5DC873F8, 0x2000) = 1073742047 0

sysctlbyname(kern.bootargs, 0xD, 0x7FFEE00DE330, 0x7FFEE00DE328, 0x0) = 0 0

issetugid(0x0, 0x0, 0x0) = 0 0

ioctl(0x2, 0x4004667A, 0x7FFEE00DDB14) = 0 0

mprotect(0x10FBB7000, 0x1000, 0x0) = 0 0

mprotect(0x10FBBC000, 0x1000, 0x0) = 0 0

mprotect(0x10FBBD000, 0x1000, 0x0) = 0 0

mprotect(0x10FBC2000, 0x1000, 0x0) = 0 0

mprotect(0x10FBB5000, 0x90, 0x1) = 0 0

mprotect(0x10FB25000, 0x1000, 0x1) = 0 0

mprotect(0x10FBB5000, 0x90, 0x3) = 0 0

mprotect(0x10FBB5000, 0x90, 0x1) = 0 0

getentropy(0x7FFEE00DDC60, 0x20, 0x0) = 0 0

getpid(0x0, 0x0, 0x0) = 55066 0

stat64("/AppleInternal\0", 0x7FFEE00DE7A0, 0x0) = -1 Err#2

csops(0xD71A, 0x7, 0x7FFEE00DE2D0) = -1 Err#22

proc\_info(0x2, 0xD71A, 0xD) = 64 0

csops(0xD71A, 0x7, 0x7FFEE00DDB20) = -1 Err#22

socketpair(0x1, 0x1, 0x0) = 0 0

fcntl(0x3, 0x2, 0x1) = 0 0

fcntl(0x4, 0x2, 0x1) = 0 0

fcntl(0x3, 0x3, 0x0) = 2 0

fcntl(0x3, 0x4, 0x6) = 0 0

fcntl(0x4, 0x3, 0x0) = 2 0

fcntl(0x4, 0x4, 0x6) = 0 0

open("/dev/urandom\0", 0x1000000, 0x0) = 5 0

socketpair(0x1, 0x1, 0x0) = 0 0

fcntl(0x6, 0x2, 0x1) = 0 0

fcntl(0x7, 0x2, 0x1) = 0 0

fcntl(0x6, 0x3, 0x0) = 2 0

fcntl(0x6, 0x4, 0x6) = 0 0

fcntl(0x7, 0x3, 0x0) = 2 0

fcntl(0x7, 0x4, 0x6) = 0 0

kqueue(0x0, 0x0, 0x0) = 8 0

kevent(0x8, 0x7FFEE00DF860, 0x1) = 0 0

bsdthread\_create(0x10FB633BA, 0x7FCF8FC02A20, 0x700000930000) = 9633792 0

thread\_selfid(0x0, 0x0, 0x0) = 195124 0

proc\_info(0x5, 0xD71A, 0x2) = 0 0

\_\_pthread\_sigmask(0x1, 0x70000092FEF4, 0x0) = 0 0

socketpair(0x1, 0x1, 0x0) = 0 0

fcntl(0x9, 0x2, 0x1) = 0 0

fcntl(0xA, 0x2, 0x1) = 0 0

fcntl(0x9, 0x3, 0x0) = 2 0

fcntl(0x9, 0x4, 0x6) = 0 0

fcntl(0xA, 0x3, 0x0) = 2 0

fcntl(0xA, 0x4, 0x6) = 0 0

kqueue(0x0, 0x0, 0x0) = 11 0

kevent(0xB, 0x7FFEE00DF860, 0x1) = 0 0

bsdthread\_create(0x10FB633BA, 0x7FCF8FC02FF0, 0x7000009B3000) = 10170368 0

proc\_info(0x5, 0xD71A, 0x2) = 0 0

thread\_selfid(0x0, 0x0, 0x0) = 195125 0

socketpair(0x1, 0x1, 0x0) = 0 0

\_\_pthread\_sigmask(0x1, 0x7000009B2EF4, 0x0) = 0 0

fcntl(0xC, 0x2, 0x1) = 0 0

fcntl(0xD, 0x2, 0x1) = 0 0

fcntl(0xC, 0x3, 0x0) = 2 0

fcntl(0xC, 0x4, 0x6) = 0 0

fcntl(0xD, 0x3, 0x0) = 2 0

fcntl(0xD, 0x4, 0x6) = 0 0

getrlimit(0x1008, 0x7FFEE00DF6A0, 0x0) = 0 0

fstat64(0x1, 0x7FFEE00DF688, 0x0) = 0 0

ioctl(0x1, 0x4004667A, 0x7FFEE00DF6D4) = 0 0

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

fstat64(0x0, 0x7FFEE00DF2C8, 0x0) = 0 0

ioctl(0x0, 0x4004667A, 0x7FFEE00DF314) = 0 0

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

4041

enter:

1) to insert money

2) to get money

3) to transfer money to another client

4) to show cash

-1) to exit

dtrace: error on enabled probe ID 2175 (ID 945: syscall::read\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

poll(0x7FFEE00DF1A0, 0x1, 0x0) = 0 0

sendto(0x9, 0x7FFEE00DF117, 0x1) = 1 0

kevent(0xB, 0x0, 0x0) = 1 0

sendto(0xC, 0x7FFEE00DF117, 0x1) = 1 0

poll(0x7000009B0DA0, 0x1, 0x0) = 1 0

recvfrom(0xA, 0x7000009B0DA7, 0x1) = 1 0

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

shm\_open(0x7FFF5DC7ACD8, 0x0, 0x0) = 14 0

mmap(0x0, 0x1000, 0x1, 0x1, 0xE, 0x0) = 0x10FBC3000 0

close\_nocancel(0xE) = 0 0

open\_nocancel("/etc/.mdns\_debug\0", 0x0, 0x0) = -1 Err#2

dtrace: error on enabled probe ID 2198 (ID 557: syscall::sysctl:return): invalid kernel access in action #10 at DIF offset 28

dtrace: error on enabled probe ID 2198 (ID 557: syscall::sysctl:return): invalid kernel access in action #10 at DIF offset 28

kqueue(0x0, 0x0, 0x0) = 14 0

kevent(0xE, 0x7000009B05B8, 0x1) = 0 0

socket(0x1, 0x1, 0x0) = 15 0

setsockopt(0xF, 0xFFFF, 0x1022) = 0 0

setsockopt(0xF, 0xFFFF, 0x1100) = 0 0

connect\_nocancel(0xF, 0x7000009B01C0, 0x6A) = 0 0

sendto\_nocancel(0xF, 0x7FCF8FF00160, 0x1C) = 28 0

select\_nocancel(0x10, 0x7000009B01B0, 0x0) = 1 0

recvfrom\_nocancel(0xF, 0x7000009B0180, 0x4) = 4 0

socketpair(0x1, 0x1, 0x0) = 0 0

setsockopt(0x10, 0xFFFF, 0x1100) = 0 0

sendto\_nocancel(0xF, 0x7FCF8FF001F0, 0x32) = 50 0

sendmsg\_nocancel(0xF, 0x7000009B00E0, 0x0) = 1 0

close\_nocancel(0x11) = 0 0

select\_nocancel(0x11, 0x7000009B00E0, 0x0) = 1 0

recvfrom\_nocancel(0x10, 0x7000009B00B0, 0x4) = 4 0

close\_nocancel(0x10) = 0 0

kevent(0xE, 0x7000009B05B8, 0x1) = 0 0

kevent(0xE, 0x0, 0x0) = 1 0

kevent(0xE, 0x0, 0x0) = 1 0

recvfrom\_nocancel(0xF, 0x7000009B0240, 0x1C) = 28 0

recvfrom\_nocancel(0xF, 0x7FCF8FF00230, 0x25) = 37 0

select\_nocancel(0x10, 0x7000009B01B0, 0x0) = 0 0

kevent(0xE, 0x7000009AFC08, 0x1) = 0 0

sendto\_nocancel(0xF, 0x7FCF8FF00200, 0x1C) = 28 0

close\_nocancel(0xE) = 0 0

socket(0x2, 0x1, 0x6) = 14 0

fcntl(0xE, 0x2, 0x1) = 0 0

setsockopt(0xE, 0xFFFF, 0x1022) = 0 0

fcntl(0xE, 0x3, 0x0) = 2 0

fcntl(0xE, 0x4, 0x6) = 0 0

connect(0xE, 0x7FCF8FE003B0, 0x10) = -1 Err#36

kevent(0xB, 0x7000009B0D60, 0x1) = 0 0

poll(0x7000009B0DA0, 0x1, 0x0) = 0 0

kevent(0xB, 0x0, 0x0) = 1 0

kevent(0xB, 0x7000009B0E00, 0x1) = 0 0

getsockopt(0xE, 0xFFFF, 0x1007) = 0 0

setsockopt(0xE, 0x6, 0x1) = 0 0

fcntl(0xE, 0x3, 0x0) = 6 0

fcntl(0xE, 0x4, 0x6) = 0 0

getpeername(0xE, 0x7000009B0D80, 0x7000009B096C) = 0 0

sendto(0x9, 0x7000009B0D57, 0x1) = 1 0

kevent(0xB, 0x0, 0x0) = 1 0

poll(0x7000009B0DA0, 0x1, 0x0) = 1 0

recvfrom(0xA, 0x7000009B0DA7, 0x1) = 1 0

kevent(0xB, 0x7000009B0C80, 0x1) = 0 0

kevent(0xB, 0x7000009B0C80, 0x1) = 0 0

recvfrom(0xE, 0x7FCF918000F0, 0xC) = -1 Err#35

poll(0x7000009B0DA0, 0x1, 0x0) = 0 0

kevent(0xB, 0x0, 0x0) = 1 0

sendto(0xE, 0x7FCF91800130, 0xA) = 10 0

kevent(0xB, 0x7000009B0E20, 0x1) = 0 0

kevent(0xB, 0x0, 0x0) = 1 0

recvfrom(0xE, 0x7FCF918000F0, 0xC) = 10 0

kevent(0xB, 0x7000009B0D90, 0x1) = 0 0

recvfrom(0xE, 0x7FCF918000FA, 0x2) = -1 Err#35

kevent(0xB, 0x0, 0x0) = 1 0

sendto(0xE, 0x7FCF9180013A, 0x1) = 1 0

kevent(0xB, 0x7000009B0E20, 0x1) = 0 0

kevent(0xB, 0x0, 0x0) = 1 0

recvfrom(0xE, 0x7FCF918000FA, 0x2) = 1 0

kevent(0xB, 0x7000009B0D90, 0x1) = 0 0

recvfrom(0xE, 0x7FCF918000FB, 0x35) = -1 Err#35

kevent(0xB, 0x0, 0x0) = 1 0

sendto(0xE, 0x7FCF9180013B, 0x35) = 53 0

kevent(0xB, 0x7000009B0E20, 0x1) = 0 0

kevent(0xB, 0x0, 0x0) = 1 0

recvfrom(0xE, 0x7FCF918000FB, 0x35) = 53 0

kevent(0xB, 0x7000009B0DE0, 0x1) = 0 0

recvfrom(0xE, 0x7FCF91003208, 0x2000) = -1 Err#35

kevent(0xB, 0x0, 0x0) = 1 0

sendto(0xE, 0x7FCF91001200, 0x28) = 40 0

kevent(0xB, 0x0, 0x0) = 2 0

kevent(0xB, 0x7000009B0E20, 0x1) = 0 0

recvfrom(0xE, 0x7FCF91003208, 0x2000) = 27 0

kevent(0xB, 0x7000009B0DC0, 0x1) = 0 0

kevent(0xB, 0x7000009B0DB0, 0x1) = 0 0

1

enter sum of money: dtrace: error on enabled probe ID 2175 (ID 945: syscall::read\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

220

successful inserting money

dtrace: error on enabled probe ID 2175 (ID 945: syscall::read\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

poll(0x7FFEE00DF760, 0x1, 0x0) = 1 0

recvfrom(0xD, 0x7FFEE00DF767, 0x1) = 1 0

poll(0x7FFEE00DF760, 0x1, 0x0) = 0 0

sendto(0x9, 0x7FFEE00DF697, 0x1) = 1 0

kevent(0xB, 0x0, 0x0) = 1 0

poll(0x7000009B0DA0, 0x1, 0x0) = 1 0

recvfrom(0xA, 0x7000009B0DA7, 0x1) = 1 0

kevent(0xB, 0x7000009B0D90, 0x1) = 0 0

sendto(0xC, 0x7000009B0C17, 0x1) = 1 0

poll(0x7FFEE00DF760, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

recvfrom(0xD, 0x7FFEE00DF767, 0x1) = 1 0

poll(0x7FFEE00DF760, 0x1, 0x0) = 0 0

sendto(0xE, 0x7FCF91001200, 0x123) = 291 0

poll(0x7000009B0DA0, 0x1, 0x0) = 0 0

kevent(0xB, 0x0, 0x0) = 1 0

kevent(0xB, 0x7000009B0E20, 0x1) = 0 0

kevent(0xB, 0x0, 0x0) = 1 0

recvfrom(0xE, 0x7FCF91003208, 0x2000) = 291 0

sendto(0xC, 0x7000009B0D57, 0x1) = 1 0

poll(0x7FFEE00DF760, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

recvfrom(0xD, 0x7FFEE00DF767, 0x1) = 1 0

poll(0x7FFEE00DF760, 0x1, 0x0) = 0 0

sendto(0x9, 0x7FFEE00DF697, 0x1) = 1 0

kevent(0xB, 0x0, 0x0) = 1 0

poll(0x7000009B0DA0, 0x1, 0x0) = 1 0

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

recvfrom(0xA, 0x7000009B0DA7, 0x1) = 1 0

poll(0x7000009B0DA0, 0x1, 0x0) = 0 0

-1

exit client

dtrace: error on enabled probe ID 2175 (ID 945: syscall::read\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

poll(0x7FFEE00DF760, 0x1, 0x0) = 0 0

sendto(0x9, 0x7FFEE00DF697, 0x1) = 1 0

kevent(0xB, 0x0, 0x0) = 1 0

poll(0x7000009B0DA0, 0x1, 0x0) = 1 0

recvfrom(0xA, 0x7000009B0DA7, 0x1) = 1 0

kevent(0xB, 0x7000009B0D90, 0x1) = 0 0

sendto(0xE, 0x7FCF91001200, 0x123) = 291 0

poll(0x7000009B0DA0, 0x1, 0x0) = 0 0

kevent(0xB, 0x0, 0x0) = 1 0

kevent(0xB, 0x7000009B0E20, 0x1) = 0 0

kevent(0xB, 0x0, 0x0) = 1 0

recvfrom(0xE, 0x7FCF91003208, 0x2000) = 291 0

sendto(0xC, 0x7000009B0D57, 0x1) = 1 0

poll(0x7FFEE00DF760, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

recvfrom(0xD, 0x7FFEE00DF767, 0x1) = 1 0

poll(0x7FFEE00DF760, 0x1, 0x0) = 0 0

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

sendto(0x6, 0x7FFEE00DF857, 0x1) = 1 0

sendto(0xC, 0x7FFEE00DF757, 0x1) = 1 0

kevent(0x8, 0x0, 0x0) = 1 0

poll(0x70000092DD60, 0x1, 0x0) = 1 0

recvfrom(0x7, 0x70000092DD67, 0x1) = 1 0

kevent(0x8, 0x70000092DD20, 0x1) = 0 0

sendto(0x9, 0x70000092DBD7, 0x1) = 1 0

poll(0x70000092DD60, 0x1, 0x0) = 0 0

kevent(0x8, 0x0, 0x0) = 1 0

poll(0x70000092DD60, 0x1, 0x0) = 1 0

recvfrom(0xD, 0x70000092DD67, 0x1) = 1 0

poll(0x70000092DD60, 0x1, 0x0) = 0 0

kevent(0xB, 0x0, 0x0) = 1 0

poll(0x7000009B0DA0, 0x1, 0x0) = 1 0

recvfrom(0xA, 0x7000009B0DA7, 0x1) = 1 0

kevent(0xB, 0x7000009B0D90, 0x1) = 0 0

sendto(0xC, 0x7000009B0BD7, 0x1) = 1 0

kevent(0xB, 0x7000009B0D80, 0x1) = 0 0

poll(0x7000009B0DA0, 0x1, 0x0) = 0 0

kevent(0x8, 0x0, 0x0) = 1 0

poll(0x70000092DD60, 0x1, 0x0) = 1 0

recvfrom(0xD, 0x70000092DD67, 0x1) = 1 0

sendto(0x9, 0x70000092DC17, 0x1) = 1 0

poll(0x70000092DD60, 0x1, 0x0) = 0 0

kevent(0xB, 0x0, 0x0) = 1 0

poll(0x7000009B0DA0, 0x1, 0x0) = 1 0

recvfrom(0xA, 0x7000009B0DA7, 0x1) = 1 0

sendto(0xC, 0x7000009B0BD7, 0x1) = 1 0

kevent(0xB, 0x7000009B0C20, 0x1) = 0 0

kevent(0x8, 0x0, 0x0) = 1 0

close(0xE) = 0 0

poll(0x70000092DD60, 0x1, 0x0) = 1 0

recvfrom(0xD, 0x70000092DD67, 0x1) = 1 0

poll(0x7000009B0DA0, 0x1, 0x0) = 0 0

poll(0x70000092DD60, 0x1, 0x0) = 0 0

kevent(0x8, 0x70000092DDE0, 0x1) = 0 0

sendto(0x6, 0x70000092DD17, 0x1) = 1 0

close(0xC) = 0 0

close(0xD) = 0 0

kevent(0x8, 0x0, 0x0) = 1 0

poll(0x70000092DD60, 0x1, 0x0) = 1 0

recvfrom(0x7, 0x70000092DD67, 0x1) = 1 0

sendto(0x3, 0x70000092DC97, 0x1) = 1 0

kevent(0x8, 0x70000092DD30, 0x1) = 0 0

poll(0x70000092DD60, 0x1, 0x0) = 0 0

\_\_disable\_threadsignal(0x1, 0x0, 0x0) = 0 0

poll(0x7FFEE00DF820, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

recvfrom(0x4, 0x7FFEE00DF827, 0x1) = 1 0

sendto(0x9, 0x7FFEE00DF717, 0x1) = 1 0

kevent(0xB, 0x0, 0x0) = 1 0

poll(0x7000009B0DA0, 0x1, 0x0) = 1 0

recvfrom(0xA, 0x7000009B0DA7, 0x1) = 1 0

kevent(0xB, 0x7000009B0D60, 0x1) = 0 0

poll(0x7000009B0DA0, 0x1, 0x0) = 0 0

\_\_disable\_threadsignal(0x1, 0x0, 0x0) = 0 0

close(0xB) = 0 0

close(0x9) = 0 0

close(0xA) = 0 0

close(0x8) = 0 0

close(0x6) = 0 0

close(0x7) = 0 0

close(0x5) = 0 0

close(0x3) = 0 0

close(0x4) = 0 0

./server

0 0

MacBook-Air-MacBook-3:os6 macbookair$ sudo dtruss ./server 4041

Password:

dtrace: system integrity protection is on, some features will not be available

SYSCALL(args) = return

open("/dev/dtracehelper\0", 0x2, 0xFFFFFFFFE7638160) = 3 0

ioctl(0x3, 0x80086804, 0x7FFEE7637F70) = 0 0

close(0x3) = 0 0

access("/AppleInternal/XBS/.isChrooted\0", 0x0, 0x0) = -1 Err#2

bsdthread\_register(0x7FFF5DC87408, 0x7FFF5DC873F8, 0x2000) = 1073742047 0

sysctlbyname(kern.bootargs, 0xD, 0x7FFEE7637330, 0x7FFEE7637328, 0x0) = 0 0

issetugid(0x0, 0x0, 0x0) = 0 0

ioctl(0x2, 0x4004667A, 0x7FFEE7636B14) = 0 0

mprotect(0x10865F000, 0x1000, 0x0) = 0 0

mprotect(0x108664000, 0x1000, 0x0) = 0 0

mprotect(0x108665000, 0x1000, 0x0) = 0 0

mprotect(0x10866A000, 0x1000, 0x0) = 0 0

mprotect(0x1085CD000, 0x90, 0x1) = 0 0

mprotect(0x1085CF000, 0x1000, 0x1) = 0 0

mprotect(0x1085CD000, 0x90, 0x3) = 0 0

mprotect(0x1085CD000, 0x90, 0x1) = 0 0

getentropy(0x7FFEE7636C60, 0x20, 0x0) = 0 0

getpid(0x0, 0x0, 0x0) = 55021 0

stat64("/AppleInternal\0", 0x7FFEE76377A0, 0x0) = -1 Err#2

csops(0xD6ED, 0x7, 0x7FFEE76372D0) = -1 Err#22

proc\_info(0x2, 0xD6ED, 0xD) = 64 0

csops(0xD6ED, 0x7, 0x7FFEE7636B20) = -1 Err#22

getrlimit(0x1008, 0x7FFEE7638510, 0x0) = 0 0

open\_nocancel("file.txt\0", 0x0, 0x1B6) = 3 0

fstat64(0x3, 0x7FFEE7637F68, 0x0) = 0 0

dtrace: error on enabled probe ID 2175 (ID 945: syscall::read\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

lstat64("file.txt\0", 0x7FFEE76385A8, 0x0) = 0 0

unlink("file.txt\0", 0x0, 0x0) = 0 0

socketpair(0x1, 0x1, 0x0) = 0 0

fcntl(0x4, 0x2, 0x1) = 0 0

fcntl(0x5, 0x2, 0x1) = 0 0

fcntl(0x4, 0x3, 0x0) = 2 0

fcntl(0x4, 0x4, 0x6) = 0 0

fcntl(0x5, 0x3, 0x0) = 2 0

fcntl(0x5, 0x4, 0x6) = 0 0

open("/dev/urandom\0", 0x1000000, 0x0) = 6 0

socketpair(0x1, 0x1, 0x0) = 0 0

fcntl(0x7, 0x2, 0x1) = 0 0

fcntl(0x8, 0x2, 0x1) = 0 0

fcntl(0x7, 0x3, 0x0) = 2 0

fcntl(0x7, 0x4, 0x6) = 0 0

fcntl(0x8, 0x3, 0x0) = 2 0

fcntl(0x8, 0x4, 0x6) = 0 0

kqueue(0x0, 0x0, 0x0) = 9 0

kevent(0x9, 0x7FFEE7638500, 0x1) = 0 0

bsdthread\_create(0x10860D3BA, 0x7F9714402A20, 0x700009B04000) = 162545664 0

thread\_selfid(0x0, 0x0, 0x0) = 194857 0

proc\_info(0x5, 0xD6ED, 0x2) = 0 0

\_\_pthread\_sigmask(0x1, 0x700009B03EF4, 0x0) = 0 0

socketpair(0x1, 0x1, 0x0) = 0 0

fcntl(0xA, 0x2, 0x1) = 0 0

fcntl(0xB, 0x2, 0x1) = 0 0

fcntl(0xA, 0x3, 0x0) = 2 0

fcntl(0xA, 0x4, 0x6) = 0 0

fcntl(0xB, 0x3, 0x0) = 2 0

fcntl(0xB, 0x4, 0x6) = 0 0

kqueue(0x0, 0x0, 0x0) = 12 0

kevent(0xC, 0x7FFEE7638500, 0x1) = 0 0

bsdthread\_create(0x10860D3BA, 0x7F9714402FF0, 0x700009B87000) = 163082240 0

proc\_info(0x5, 0xD6ED, 0x2) = 0 0

thread\_selfid(0x0, 0x0, 0x0) = 194858 0

\_\_pthread\_sigmask(0x1, 0x700009B86EF4, 0x0) = 0 0

socketpair(0x1, 0x1, 0x0) = 0 0

fcntl(0xD, 0x2, 0x1) = 0 0

fcntl(0xE, 0x2, 0x1) = 0 0

fcntl(0xD, 0x3, 0x0) = 2 0

fcntl(0xD, 0x4, 0x6) = 0 0

fcntl(0xE, 0x3, 0x0) = 2 0

fcntl(0xE, 0x4, 0x6) = 0 0

poll(0x7FFEE7638160, 0x1, 0x0) = 0 0

shm\_open(0x7FFF5DC7ACD8, 0x0, 0x0) = 15 0

mmap(0x0, 0x1000, 0x1, 0x1, 0xF, 0x0) = 0x10866B000 0

close\_nocancel(0xF) = 0 0

open\_nocancel("/etc/.mdns\_debug\0", 0x0, 0x0) = -1 Err#2

socket(0x2, 0x1, 0x6) = 15 0

fcntl(0xF, 0x2, 0x1) = 0 0

setsockopt(0xF, 0xFFFF, 0x1022) = 0 0

fcntl(0xF, 0x2, 0x1) = 0 0

setsockopt(0xF, 0xFFFF, 0x4) = 0 0

bind(0xF, 0x7F9714403490, 0x10) = 0 0

listen(0xF, 0x64, 0x0) = 0 0

getsockname(0xF, 0x7FFEE76381F8, 0x7FFEE76381F4) = 0 0

sendto(0xA, 0x7FFEE76380D7, 0x1) = 1 0

kevent(0xC, 0x0, 0x0) = 1 0

sendto(0xD, 0x7FFEE7638117, 0x1) = 1 0

poll(0x700009B84DA0, 0x1, 0x0) = 1 0

sigaction(0x2, 0x7FFEE76385F8, 0x7FFEE7638620) = 0 0

recvfrom(0xB, 0x700009B84DA7, 0x1) = 1 0

poll(0x7FFEE76384A0, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

kevent(0xC, 0x700009B84D90, 0x1) = 0 0

recvfrom(0xE, 0x7FFEE76384A7, 0x1) = 1 0

poll(0x700009B84DA0, 0x1, 0x0) = 0 0

poll(0x7FFEE76384A0, 0x1, 0x0) = 0 0

kevent(0xC, 0x0, 0x0) = 1 0

accept(0xF, 0x700009B84DA0, 0x700009B84D9C) = 16 0

fcntl(0x10, 0x2, 0x1) = 0 0

setsockopt(0x10, 0xFFFF, 0x1022) = 0 0

setsockopt(0x10, 0x6, 0x1) = 0 0

fcntl(0x10, 0x3, 0x0) = 2 0

fcntl(0x10, 0x4, 0x6) = 0 0

getpeername(0x10, 0x700009B84D70, 0x700009B8495C) = 0 0

sendto(0xA, 0x700009B84D17, 0x1) = 1 0

kevent(0xC, 0x0, 0x0) = 1 0

poll(0x700009B84DA0, 0x1, 0x0) = 1 0

recvfrom(0xB, 0x700009B84DA7, 0x1) = 1 0

sendto(0xD, 0x700009B84C57, 0x1) = 1 0

poll(0x7FFEE76384A0, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

kevent(0xC, 0x700009B84C80, 0x1) = 0 0

recvfrom(0xE, 0x7FFEE76384A7, 0x1) = 1 0

kevent(0xC, 0x700009B84C80, 0x1) = 0 0

recvfrom(0x10, 0x7F97150000F0, 0xC) = 10 0

recvfrom(0x10, 0x7F97150000FA, 0x2) = -1 Err#35

poll(0x7FFEE76384A0, 0x1, 0x0) = 0 0

poll(0x700009B84DA0, 0x1, 0x0) = 0 0

kevent(0xC, 0x0, 0x0) = 1 0

sendto(0x10, 0x7F9715000130, 0xB) = 11 0

kevent(0xC, 0x700009B84E20, 0x1) = 0 0

kevent(0xC, 0x0, 0x0) = 1 0

recvfrom(0x10, 0x7F97150000FA, 0x2) = 2 0

kevent(0xC, 0x700009B84D90, 0x1) = 0 0

recvfrom(0x10, 0x7F97150000FC, 0x34) = 52 0

recvfrom(0x10, 0x7F971500AA08, 0x2000) = -1 Err#35

kevent(0xC, 0x0, 0x0) = 1 0

sendto(0x10, 0x7F971500013B, 0x35) = 53 0

kevent(0xC, 0x0, 0x0) = 1 0

sendto(0x10, 0x7F9715008A00, 0x1B) = 27 0

kevent(0xC, 0x0, 0x0) = 1 0

kevent(0xC, 0x700009B84E20, 0x1) = 0 0

kevent(0xC, 0x0, 0x0) = 1 0

recvfrom(0x10, 0x7F971500AA08, 0x2000) = 40 0

sendto(0xD, 0x700009B84C97, 0x1) = 1 0

poll(0x7FFEE76384A0, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

recvfrom(0xE, 0x7FFEE76384A7, 0x1) = 1 0

kevent(0xC, 0x700009B84DC0, 0x1) = 0 0

sendto(0xA, 0x7FFEE7638297, 0x1) = 1 0

kevent(0xC, 0x700009B84DB0, 0x1) = 0 0

kevent(0xC, 0x0, 0x0) = 1 0

poll(0x7FFEE76384A0, 0x1, 0x0) = 0 0

poll(0x700009B84DA0, 0x1, 0x0) = 1 0

recvfrom(0xB, 0x700009B84DA7, 0x1) = 1 0

poll(0x700009B84DA0, 0x1, 0x0) = 0 0

kevent(0xC, 0x0, 0x0) = 1 0

accept(0xF, 0x700009B84DA0, 0x700009B84D9C) = 17 0

fcntl(0x11, 0x2, 0x1) = 0 0

setsockopt(0x11, 0xFFFF, 0x1022) = 0 0

setsockopt(0x11, 0x6, 0x1) = 0 0

fcntl(0x11, 0x3, 0x0) = 2 0

fcntl(0x11, 0x4, 0x6) = 0 0

getpeername(0x11, 0x700009B84D70, 0x700009B8495C) = 0 0

sendto(0xA, 0x700009B84D17, 0x1) = 1 0

kevent(0xC, 0x0, 0x0) = 1 0

poll(0x700009B84DA0, 0x1, 0x0) = 1 0

recvfrom(0xB, 0x700009B84DA7, 0x1) = 1 0

sendto(0xD, 0x700009B84C57, 0x1) = 1 0

poll(0x7FFEE76384A0, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

recvfrom(0xE, 0x7FFEE76384A7, 0x1) = 1 0

kevent(0xC, 0x700009B84C80, 0x1) = 0 0

poll(0x7FFEE76384A0, 0x1, 0x0) = 0 0

kevent(0xC, 0x700009B84C80, 0x1) = 0 0

recvfrom(0x11, 0x7F971500F2F0, 0xC) = -1 Err#35

poll(0x700009B84DA0, 0x1, 0x0) = 0 0

kevent(0xC, 0x0, 0x0) = 1 0

sendto(0x11, 0x7F971500F330, 0xA) = 10 0

kevent(0xC, 0x700009B84E20, 0x1) = 0 0

kevent(0xC, 0x0, 0x0) = 1 0

recvfrom(0x11, 0x7F971500F2F0, 0xC) = 10 0

kevent(0xC, 0x700009B84D90, 0x1) = 0 0

recvfrom(0x11, 0x7F971500F2FA, 0x2) = -1 Err#35

kevent(0xC, 0x0, 0x0) = 1 0

sendto(0x11, 0x7F971500F33A, 0x1) = 1 0

kevent(0xC, 0x700009B84E20, 0x1) = 0 0

kevent(0xC, 0x0, 0x0) = 1 0

recvfrom(0x11, 0x7F971500F2FA, 0x2) = 1 0

kevent(0xC, 0x700009B84D90, 0x1) = 0 0

recvfrom(0x11, 0x7F971500F2FB, 0x35) = -1 Err#35

kevent(0xC, 0x0, 0x0) = 1 0

sendto(0x11, 0x7F971500F33B, 0x35) = 53 0

kevent(0xC, 0x700009B84E20, 0x1) = 0 0

kevent(0xC, 0x0, 0x0) = 1 0

recvfrom(0x11, 0x7F971500F2FB, 0x35) = 53 0

kevent(0xC, 0x700009B84DE0, 0x1) = 0 0

recvfrom(0x11, 0x7F9715802008, 0x2000) = -1 Err#35

kevent(0xC, 0x0, 0x0) = 1 0

sendto(0x11, 0x7F9715800000, 0x1B) = 27 0

kevent(0xC, 0x0, 0x0) = 1 0

kevent(0xC, 0x700009B84E20, 0x1) = 0 0

kevent(0xC, 0x0, 0x0) = 1 0

recvfrom(0x11, 0x7F9715802008, 0x2000) = 40 0

sendto(0xD, 0x700009B84C97, 0x1) = 1 0

poll(0x7FFEE76384A0, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

recvfrom(0xE, 0x7FFEE76384A7, 0x1) = 1 0

kevent(0xC, 0x700009B84DC0, 0x1) = 0 0

kevent(0xC, 0x700009B84DB0, 0x1) = 0 0

sendto(0xA, 0x7FFEE7638297, 0x1) = 1 0

kevent(0xC, 0x0, 0x0) = 1 0

poll(0x7FFEE76384A0, 0x1, 0x0) = 0 0

poll(0x700009B84DA0, 0x1, 0x0) = 1 0

recvfrom(0xB, 0x700009B84DA7, 0x1) = 1 0

poll(0x700009B84DA0, 0x1, 0x0) = 0 0

0

id : 10 ; insert sum : 220

kevent(0xC, 0x0, 0x0) = 1 0

recvfrom(0x11, 0x7F9715802008, 0x2000) = 291 0

sendto(0xD, 0x700009B84D57, 0x1) = 1 0

poll(0x7FFEE76384A0, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

recvfrom(0xE, 0x7FFEE76384A7, 0x1) = 1 0

poll(0x7FFEE76384A0, 0x1, 0x0) = 0 0

sendto(0xA, 0x7FFEE76383D7, 0x1) = 1 0

kevent(0xC, 0x0, 0x0) = 1 0

poll(0x700009B84DA0, 0x1, 0x0) = 1 0

fstat64(0x1, 0x7FFEE7638328, 0x0) = 0 0

recvfrom(0xB, 0x700009B84DA7, 0x1) = 1 0

ioctl(0x1, 0x4004667A, 0x7FFEE7638374) = 0 0

poll(0x700009B84DA0, 0x1, 0x0) = 0 0

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

poll(0x7FFEE76384A0, 0x1, 0x0) = 0 0

sendto(0xA, 0x7FFEE7638457, 0x1) = 1 0

kevent(0xC, 0x0, 0x0) = 1 0

poll(0x700009B84DA0, 0x1, 0x0) = 1 0

recvfrom(0xB, 0x700009B84DA7, 0x1) = 1 0

kevent(0xC, 0x700009B84D90, 0x1) = 0 0

sendto(0xD, 0x700009B84C17, 0x1) = 1 0

poll(0x7FFEE76384A0, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

recvfrom(0xE, 0x7FFEE76384A7, 0x1) = 1 0

sendto(0x11, 0x7F9715800000, 0x123) = 291 0

poll(0x700009B84DA0, 0x1, 0x0) = 0 0

poll(0x7FFEE76384A0, 0x1, 0x0) = 0 0

kevent(0xC, 0x0, 0x0) = 1 0

kevent(0xC, 0x700009B84E20, 0x1) = 0 0

try delete

kek

0 1

id 10 sum 220

kevent(0xC, 0x0, 0x0) = 1 0

recvfrom(0x11, 0x7F9715802008, 0x2000) = 291 0

sendto(0xD, 0x700009B84D57, 0x1) = 1 0

poll(0x7FFEE76384A0, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

recvfrom(0xE, 0x7FFEE76384A7, 0x1) = 1 0

poll(0x7FFEE76384A0, 0x1, 0x0) = 0 0

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

poll(0x7FFEE76384A0, 0x1, 0x0) = 0 0

sendto(0xA, 0x7FFEE7638457, 0x1) = 1 0

kevent(0xC, 0x0, 0x0) = 1 0

poll(0x700009B84DA0, 0x1, 0x0) = 1 0

recvfrom(0xB, 0x700009B84DA7, 0x1) = 1 0

kevent(0xC, 0x700009B84D90, 0x1) = 0 0

sendto(0x11, 0x7F9715800000, 0x123) = 291 0

poll(0x700009B84DA0, 0x1, 0x0) = 0 0

open\_nocancel("file.txt\0", 0x20A, 0x1B6) = 18 0

kevent(0xC, 0x0, 0x0) = 1 0

kevent(0xC, 0x700009B84E20, 0x1) = 0 0

lseek(0x12, 0x0, 0x2) = 0 0

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

fstat64(0x12, 0x7FFEE7638328, 0x0) = 0 0

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

sendto(0x7, 0x7FFEE76384D7, 0x1) = 1 0

sendto(0xD, 0x7FFEE76383D7, 0x1) = 1 0

kevent(0x9, 0x0, 0x0) = 1 0

poll(0x700009B01D60, 0x1, 0x0) = 1 0

recvfrom(0x8, 0x700009B01D67, 0x1) = 1 0

kevent(0x9, 0x700009B01D20, 0x1) = 0 0

sendto(0xA, 0x700009B01BD7, 0x1) = 1 0

kevent(0xC, 0x0, 0x0) = 1 0

poll(0x700009B84DA0, 0x1, 0x0) = 1 0

poll(0x700009B01D60, 0x1, 0x0) = 0 0

kevent(0x9, 0x0, 0x0) = 1 0

poll(0x700009B01D60, 0x1, 0x0) = 1 0

recvfrom(0xE, 0x700009B01D67, 0x1) = 1 0

poll(0x700009B01D60, 0x1, 0x0) = 0 0

recvfrom(0xB, 0x700009B84DA7, 0x1) = 1 0

kevent(0xC, 0x700009B84D90, 0x1) = 0 0

sendto(0xD, 0x700009B84BD7, 0x1) = 1 0

kevent(0x9, 0x0, 0x0) = 1 0

kevent(0xC, 0x700009B84D80, 0x1) = 0 0

poll(0x700009B01D60, 0x1, 0x0) = 1 0

kevent(0xC, 0x700009B84D90, 0x1) = 0 0

kevent(0xC, 0x700009B84D80, 0x1) = 0 0

kevent(0xC, 0x700009B84D70, 0x1) = 0 0

recvfrom(0xE, 0x700009B01D67, 0x1) = 1 0

poll(0x700009B01D60, 0x1, 0x0) = 0 0

close(0xF) = 0 0

kevent(0xC, 0x700009B84CF0, 0x1) = 0 0

close(0x10) = 0 0

kevent(0xC, 0x700009B84CF0, 0x1) = 0 0

close(0x11) = 0 0

sendto(0xD, 0x700009B84CD7, 0x1) = 1 0

poll(0x700009B84DA0, 0x1, 0x0) = 0 0

kevent(0x9, 0x0, 0x0) = 1 0

poll(0x700009B01D60, 0x1, 0x0) = 1 0

recvfrom(0xE, 0x700009B01D67, 0x1) = 1 0

poll(0x700009B01D60, 0x1, 0x0) = 0 0

kevent(0x9, 0x700009B01DE0, 0x1) = 0 0

sendto(0x7, 0x700009B01D17, 0x1) = 1 0

close(0xD) = 0 0

close(0xE) = 0 0

kevent(0x9, 0x0, 0x0) = 1 0

poll(0x700009B01D60, 0x1, 0x0) = 1 0

recvfrom(0x8, 0x700009B01D67, 0x1) = 1 0

sendto(0x4, 0x700009B01C97, 0x1) = 1 0

kevent(0x9, 0x700009B01D30, 0x1) = 0 0

poll(0x700009B01D60, 0x1, 0x0) = 0 0

poll(0x7FFEE76384A0, 0x1, 0xFFFFFFFFFFFFFFFF) = 1 0

\_\_disable\_threadsignal(0x1, 0x0, 0x0) = 0 0

recvfrom(0x5, 0x7FFEE76384A7, 0x1) = 1 0

sendto(0xA, 0x7FFEE7638397, 0x1) = 1 0

kevent(0xC, 0x0, 0x0) = 1 0

poll(0x700009B84DA0, 0x1, 0x0) = 1 0

recvfrom(0xB, 0x700009B84DA7, 0x1) = 1 0

kevent(0xC, 0x700009B84D60, 0x1) = 0 0

poll(0x700009B84DA0, 0x1, 0x0) = 0 0

\_\_disable\_threadsignal(0x1, 0x0, 0x0) = 0 0

ulock\_wake(0x1000002, 0x700009B87034, 0x0) = 0 0

ulock\_wait(0x1020002, 0x700009B87034, 0x1103) = 0 0

close(0xC) = 0 0

close(0xA) = 0 0

close(0xB) = 0 0

close(0x9) = 0 0

close(0x7) = 0 0

close(0x8) = 0 0

close(0x6) = 0 0

close(0x4) = 0 0

close(0x5) = 0 0

dtrace: error on enabled probe ID 2173 (ID 947: syscall::write\_nocancel:return): invalid kernel access in action #12 at DIF offset 68

Вывод:

Я написал довольно простую программу типа клиент - сервер с помощью zmq , что упростило написание . Без этого пришлось бы использовать много всего ненужного , а так всю "грязную" работу выполняет сервер сообщений .

Данная работа останется в памяти и возможно именно zmq я буду использовать , если понадобится решить подобного рода задачи .