****Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работ №5 по курсу**

**«Операционные системы»**

Группа: М80 – 207Б – 18

Студент: Син Денис Дмитриевич

Преподаватель: Миронов Евгений Сергеевич

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

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2019.

**Содержание**

1. Постановка задачи
2. Общие сведения о программе
3. Основные файлы программы
4. Демонстрация работы программы
5. Вывод

**Постановка задачи**.

Требуется создать динамическую библиотеку, которая реализует определенный функционал. Далее использовать данную библиотеку 2-мя способами:

1. Во время компиляции (на этапе «линковки»/linking)

2. Во время исполнения программы, подгрузив библиотеку в память с помощью системных вызовов

В конечном итоге, программа должна состоять из следующих частей

* Динамическая библиотека, реализующая заданных вариантом интерфейс;
* Тестовая программа, которая используют библиотеку, используя знания полученные на этапе компиляции;
* Тестовая программа, которая использует библиотеку, используя только местоположение динамической библиотеки и ее интерфейс.

Провести анализ между обоими типами использования библиотеки.

**Общие сведения о программе**

Дерево проекта:

**.**

**├── Makefile**

**├── ct\_main.c**

**├── rt\_main.c**

**└── src**

**├── str\_array.c**

**└── str\_array.h**

В файле str\_array.c находится реализация массива. В файле str\_array.h находится объявление структуры массива. В файлах ct\_main.c и rt\_main.c находятся реализации программ с разным типом использования динамической библиотеки. В файле Makefile находятся правила для сборки динамической библиотеки и компилирования исполняемых файлов.

**Основные файлы программы**.

**Файл rt\_main.c**

#include <stdlib.h>

#include <stdio.h>

#include <dlfcn.h>

#include "src/str\_array.h"

int main(void)

{

void \*library\_handler = dlopen("libstr\_array.so", RTLD\_LAZY);

if (!library\_handler) {

fprintf(stderr,"dlopen() error: %s\n", dlerror());

return 1;

}

str\_array\_t\*(\*str\_array\_create)(size\_t size) = dlsym(library\_handler, "sr\_array\_create");

char\*(\*str\_array\_get\_item)(str\_array\_t\* arr, size\_t idx) = dlsym(library\_handler, "str\_array\_get\_item");

void(\*str\_array\_set\_item)(str\_array\_t\* arr, size\_t idx, const char\* str) = dlsym(library\_handler, "str\_array\_set\_item");

void(\*str\_array\_destroy)(str\_array\_t \*\*arr) = dlsym(library\_handler, "str\_array\_destroy");

void(\*str\_array\_print)(str\_array\_t\* arr) = dlsym(library\_handler, "str\_array\_print");

if (str\_array\_create == NULL || str\_array\_set\_item == NULL ||

str\_array\_destroy == NULL || str\_array\_print == NULL || str\_array\_set\_item == NULL) {

char\* err = dlerror();

printf("%s\n", err);

return 1;

}

str\_array\_t\* arr = (\*str\_array\_create)(10);

(\*str\_array\_set\_item)(arr, 0, "e23e2efffaaabb232323ab");

(\*str\_array\_set\_item)(arr, 1, "e23e9879877fefef879793ab");

(\*str\_array\_set\_item)(arr, 2, "e2988389798ef8ef98e7ab7aba9b");

(\*str\_array\_set\_item)(arr, 3, "efef56ef5e6f5e8232738abccc32");

(\*str\_array\_set\_item)(arr, 4, "e9898987797fef7ef7e9fe7ababab32");

(\*str\_array\_set\_item)(arr, 5, "67667766fefefeabca787878b7b7b7");

(\*str\_array\_set\_item)(arr, 6, "e7ba6767e76e767e6e7e7e67c67c6c799");

(\*str\_array\_set\_item)(arr, 7, "e42352454baceceff7837273788b");

(\*str\_array\_set\_item)(arr, 8, "e36636efef77abc7bc77778283828");

(\*str\_array\_set\_item)(arr, 9, "e2832938ebfbef838283ae23888efe323");

(\*str\_array\_get\_item)(arr, 0);

str\_array\_print(arr);

str\_array\_destroy(&arr);

dlclose(library\_handler);

return 0;

}

**Файл ct\_main.c**

#include <stdlib.h>

#include <stdio.h>

#include <dlfcn.h>

#include "src/str\_array.h"

extern str\_array\_t\* str\_array\_create(size\_t *size*);

extern void str\_array\_destroy(str\_array\_t \*\**arr*);

extern char\* str\_array\_get\_item(str\_array\_t\* *arr*, size\_t *idx*);

extern void str\_array\_set\_item(str\_array\_t\* *arr*, size\_t *idx*, const char\* *str*);

extern void str\_array\_print(str\_array\_t\* *arr*);

int main(void)

{

str\_array\_t\* arr = str\_array\_create(10);

str\_array\_set\_item(arr, 0, "e23e2efffaaabb232323ab");

str\_array\_set\_item(arr, 1, "e23e9879877fefef879793ab");

str\_array\_set\_item(arr, 2, "e2988389798ef8ef98e7ab7aba9b");

str\_array\_set\_item(arr, 3, "efef56ef5e6f5e8232738abccc32");

str\_array\_set\_item(arr, 4, "e9898987797fef7ef7e9fe7ababab32");

str\_array\_set\_item(arr, 5, "67667766fefefeabca787878b7b7b7");

str\_array\_set\_item(arr, 6, "e7ba6767e76e767e6e7e7e67c67c6c799");

str\_array\_set\_item(arr, 7, "e42352454baceceff7837273788b");

str\_array\_set\_item(arr, 8, "e36636efef77abc7bc77778283828");

str\_array\_set\_item(arr, 9, "e2832938ebfbef838283ae23888efe323");

str\_array\_print(arr);

str\_array\_destroy(&arr);

return 0;

}

**Файл str\_array.c**

#include "str\_array.h"

str\_array\_t\* str\_array\_create(size\_t *size*) {

str\_array\_t\* array = (str\_array\_t\*)malloc(sizeof(str\_array\_t));

array->data\_ = (char \*\*)malloc(sizeof(char\*) \* size);

array->size\_ = size;

for (size\_t i = 0; i < size; ++i) {

array->data\_[i] = NULL;

}

return array;

}

void str\_array\_destroy(str\_array\_t \*\**arr*) {

for (size\_t i = 0; i < (\*arr)->size\_; ++i) {

free((\*arr)->data\_[i]);

(\*arr)->data\_[i] = NULL;

}

free((\*arr)->data\_);

(\*arr)->size\_ = 0;

(\*arr)->data\_ = NULL;

arr = NULL;

}

char\* str\_array\_get\_item(str\_array\_t\* *arr*, size\_t *idx*) {

return arr->data\_[idx];

}

void str\_array\_set\_item(str\_array\_t\* *arr*, size\_t *idx*, const char\* *str*) {

free(arr->data\_[idx]);

arr->data\_[idx] = strdup(str);

}

void str\_array\_print(str\_array\_t\* *arr*) {

for (size\_t i = 0; i < arr->size\_; ++i) {

printf("%2zu: %s\n", i, arr->data\_[i]);

}

}

**Файл str\_array.h**

#ifndef STR\_ARRAY\_H

#define STR\_ARRAY\_H

#include <stdlib.h>

#include <string.h>

#include <stdio.h>

typedef struct str\_array {

char\*\* data\_;

size\_t size\_;

} str\_array\_t;

#endif

**Файл Makefile**

CC = gcc

LIB = str\_array

FLAGS = -Wall -Werror -Wextra

FILES = str\_array.c

SRC\_DIR = src

OBJ\_DIR = obj

SRCS = $(addprefix $(SRC\_DIR)/, $(FILES))

OBJS = $(addprefix $(OBJ\_DIR)/, $(FILES:.c=.o))

RT\_NAME = rt\_main

CT\_NAME = ct\_main

all: $(CT\_NAME) $(RT\_NAME)

$(CT\_NAME): lib$(LIB).so ct\_main.c

@$(CC) ct\_main.c -L. -lstr\_array -o $(CT\_NAME)

@echo $(CT\_NAME) is builded

$(RT\_NAME): lib$(LIB).so rt\_main.c

@$(CC) $(FLAGS) rt\_main.c -o $(RT\_NAME)

@echo $(RT\_NAME) is builded

lib$(LIB).so: $(OBJ\_DIR) $(OBJS)

@$(CC) -shared $(OBJS) -o lib$(LIB).so

@echo lib$(LIB).so is builded

$(OBJ\_DIR)/%.o: $(SRC\_DIR)/%.c

@$(CC) $(FLAGS) -fPIC -c $^ -o $@

$(OBJ\_DIR):

@mkdir -p $(OBJ\_DIR)

clean:

@rm -rf lib$(LIB).so

@rm -rf $(OBJ\_DIR)

@rm -rf $(RT\_NAME)

@rm -rf $(CT\_NAME)

**Демонстрация работы программы.**

~/CSinDenis/university/sem3/OperatingSystems/lab05/c(master\*) » ls

Makefile ct\_main.c rt\_main.c src

--------------------------------------------------------------------------------

~/CSinDenis/university/sem3/OperatingSystems/lab05/c(master\*) » make

libstr\_array.so is builded

ct\_main is builded

rt\_main is builded

--------------------------------------------------------------------------------

~/CSinDenis/university/sem3/OperatingSystems/lab05/c(master\*) » ls

Makefile ct\_main.c obj rt\_main.c

ct\_main libstr\_array.so rt\_main src

--------------------------------------------------------------------------------

~/CSinDenis/university/sem3/OperatingSystems/lab05/c(master\*) » ./rt\_main

0: e23e2efffaaabb232323ab

1: e23e9879877fefef879793ab

2: e2988389798ef8ef98e7ab7aba9b

3: efef56ef5e6f5e8232738abccc32

4: e9898987797fef7ef7e9fe7ababab32

5: 67667766fefefeabca787878b7b7b7

6: e7ba6767e76e767e6e7e7e67c67c6c799

7: e42352454baceceff7837273788b

8: e36636efef77abc7bc77778283828

9: e2832938ebfbef838283ae23888efe323

--------------------------------------------------------------------------------

~/CSinDenis/university/sem3/OperatingSystems/lab05/c(master\*) » ./ct\_main

0: e23e2efffaaabb232323ab

1: e23e9879877fefef879793ab

2: e2988389798ef8ef98e7ab7aba9b

3: efef56ef5e6f5e8232738abccc32

4: e9898987797fef7ef7e9fe7ababab32

5: 67667766fefefeabca787878b7b7b7

6: e7ba6767e76e767e6e7e7e67c67c6c799

7: e42352454baceceff7837273788b

8: e36636efef77abc7bc77778283828

9: e2832938ebfbef838283ae23888efe323

**Вывод**

При разработке больших проектов программисты пишут отдельные модули, каждые из которых отвечают за некоторую отдельную логику. Эти отдельные модули в языках C/C++ могут быть двух типов статические и динамические библиотеки. Статические библиотеки линкуются к программе на этапе компиляции и для каждой отдельной программы они свои. Динамические библиотеки загружаются после запуска программы. Основным преимуществом работой с динамическими библиотеками является совместное их использование несколькими программами, что позволяет экономить память. Также есть возможность собрать динамическую библиотеку, в которой функции реализованы на языке C или C++, а затем в программе, написанной на языке python вызывать данные функции из динамической библиотеки, таким образом некоторые функции будут работать быстрее, чем если бы их реализовывали в python.

**Трейсы dtruss для ct\_main**

PID/THRD SYSCALL(args) = return

42258/0xe8e61: thread\_selfid(0x0, 0x0, 0x0) = 953953 0

42258/0xe8e61: open(".\0", 0x0, 0x1) = 3 0

42258/0xe8e61: fstat64(0x3, 0x7FFEEE0A8EB0, 0x0) = 0 0

42258/0xe8e61: fcntl(0x3, 0x32, 0x7FFEEE0A9580) = 0 0

42258/0xe8e61: close(0x3) = 0 0

42258/0xe8e61: stat64("/Users/sindenis/CSinDenis/university/sem3/OperatingSystems/lab05/c\0", 0x7FFEEE0A8E20, 0x0) = 0 0

42258/0xe8e61: \_\_mac\_syscall(0x10DED2C5F, 0x5A, 0x7FFEEE0A9080) = 0 0

42258/0xe8e61: shared\_region\_check\_np(0x7FFEEE0A8EE8, 0x0, 0x0) = 0 0

42258/0xe8e61: stat64("/private/var/db/dyld/dyld\_shared\_cache\_x86\_64h\0", 0x7FFEEE0A8E30, 0x0) = 0 0

42258/0xe8e61: csrctl(0x0, 0x7FFEEE0A8BFC, 0x4) = -1 Err#1

42258/0xe8e61: stat64("/Users/sindenis/CSinDenis/university/sem3/OperatingSystems/lab05/c/./ct\_main\0", 0x7FFEEE0A9580, 0x0) = 0 0

42258/0xe8e61: getpid(0x0, 0x0, 0x0) = 42258 0

42258/0xe8e61: proc\_info(0x2, 0xA512, 0x16) = 1272 0

42258/0xe8e61: stat64("libstr\_array.so\0", 0x7FFEEE0A7E30, 0x0) = 0 0

42258/0xe8e61: stat64("libstr\_array.so\0", 0x7FFEEE0A8270, 0x0) = 0 0

42258/0xe8e61: open("libstr\_array.so\0", 0x0, 0x0) = 3 0

dtrace: error on enabled probe ID 2189 (ID 468: syscall::pread:return): invalid kernel access in action #13 at DIF offset 20

42258/0xe8e61: fcntl(0x3, 0x62, 0x7FFEEE09FA00) = 0 0

42258/0xe8e61: mmap(0x101B5E000, 0x1000, 0x5, 0x12, 0x3, 0x0) = 0x101B5E000 0

42258/0xe8e61: mmap(0x101B5F000, 0x1000, 0x3, 0x12, 0x3, 0x1000) = 0x101B5F000 0

42258/0xe8e61: mmap(0x101B60000, 0x218, 0x1, 0x12, 0x3, 0x2000) = 0x101B60000 0

42258/0xe8e61: fcntl(0x3, 0x32, 0x7FFEEE09FC90) = 0 0

42258/0xe8e61: fcntl(0x3, 0x2C, 0x7FFEEE09FC00) = 0 0

42258/0xe8e61: close(0x3) = 0 0

42258/0xe8e61: stat64("/usr/lib/libSystem.B.dylib\0", 0x7FFEEE0A8270, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libcache.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libcommonCrypto.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libcompiler\_rt.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libcopyfile.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libcorecrypto.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libdispatch.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libdyld.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libkeymgr.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/liblaunch.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libmacho.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libquarantine.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libremovefile.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_asl.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_blocks.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_c.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_configuration.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_coreservices.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_darwin.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_dnssd.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_info.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_m.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_malloc.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_networkextension.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_notify.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_sandbox.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_secinit.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_kernel.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_platform.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_pthread.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_symptoms.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libsystem\_trace.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libunwind.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/system/libxpc.dylib\0", 0x7FFEEE0A7DD0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/libobjc.A.dylib\0", 0x7FFEEE0A70D0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/libc++abi.dylib\0", 0x7FFEEE0A67B0, 0x0) = 0 0

42258/0xe8e61: stat64("/usr/lib/libc++.1.dylib\0", 0x7FFEEE0A67B0, 0x0) = 0 0

42258/0xe8e61: open("/dev/dtracehelper\0", 0x2, 0xFFFFFFFFEE0A9010) = 3 0

42258/0xe8e61: ioctl(0x3, 0x80086804, 0x7FFEEE0A8E20) = 0 0

42258/0xe8e61: close(0x3) = 0 0

42258/0xe8e61: access("/AppleInternal/XBS/.isChrooted\0", 0x0, 0x0) = -1 Err#2

42258/0xe8e61: bsdthread\_register(0x7FFF60845400, 0x7FFF608453F0, 0x2000) = 1073742047 0

42258/0xe8e61: sysctlbyname(kern.bootargs, 0xD, 0x7FFEEE0A81E0, 0x7FFEEE0A81D0, 0x0) = 0 0

42258/0xe8e61: issetugid(0x0, 0x0, 0x0) = 0 0

42258/0xe8e61: ioctl(0x2, 0x4004667A, 0x7FFEEE0A79B4) = 0 0

42258/0xe8e61: mprotect(0x101B61000, 0x1000, 0x0) = 0 0

42258/0xe8e61: mprotect(0x101B68000, 0x1000, 0x0) = 0 0

42258/0xe8e61: mprotect(0x101B69000, 0x1000, 0x0) = 0 0

42258/0xe8e61: mprotect(0x101B70000, 0x1000, 0x0) = 0 0

42258/0xe8e61: mprotect(0x101B5B000, 0x90, 0x1) = 0 0

42258/0xe8e61: mprotect(0x101B5D000, 0x1000, 0x1) = 0 0

42258/0xe8e61: mprotect(0x101B5B000, 0x90, 0x3) = 0 0

42258/0xe8e61: mprotect(0x101B5B000, 0x90, 0x1) = 0 0

42258/0xe8e61: getentropy(0x7FFEEE0A7B00, 0x20, 0x0) = 0 0

42258/0xe8e61: getpid(0x0, 0x0, 0x0) = 42258 0

42258/0xe8e61: stat64("/AppleInternal\0", 0x7FFEEE0A8650, 0x0) = -1 Err#2

42258/0xe8e61: csops(0xA512, 0x7, 0x7FFEEE0A8180) = -1 Err#22

42258/0xe8e61: proc\_info(0x2, 0xA512, 0xD) = 64 0

42258/0xe8e61: csops(0xA512, 0x7, 0x7FFEEE0A79C0) = -1 Err#22

42258/0xe8e61: getrlimit(0x1008, 0x7FFEEE0A9700, 0x0) = 0 0

42258/0xe8e61: fstat64(0x1, 0x7FFEEE0A96E8, 0x0) = 0 0

42258/0xe8e61: ioctl(0x1, 0x4004667A, 0x7FFEEE0A9734) = 0 0

**Трейсы dtruss для rt\_main**

PID/THRD SYSCALL(args) = return

42321/0xe91bc: thread\_selfid(0x0, 0x0, 0x0) = 954812 0

42321/0xe91bc: open(".\0", 0x0, 0x1) = 3 0

42321/0xe91bc: fstat64(0x3, 0x7FFEE177AEB0, 0x0) = 0 0

42321/0xe91bc: fcntl(0x3, 0x32, 0x7FFEE177B580) = 0 0

42321/0xe91bc: close(0x3) = 0 0

42321/0xe91bc: stat64("/Users/sindenis/CSinDenis/university/sem3/OperatingSystems/lab05/c\0", 0x7FFEE177AE20, 0x0) = 0 0

42321/0xe91bc: \_\_mac\_syscall(0x10F672C5F, 0x5A, 0x7FFEE177B080) = 0 0

42321/0xe91bc: shared\_region\_check\_np(0x7FFEE177AEE8, 0x0, 0x0) = 0 0

42321/0xe91bc: stat64("/private/var/db/dyld/dyld\_shared\_cache\_x86\_64h\0", 0x7FFEE177AE30, 0x0) = 0 0

42321/0xe91bc: csrctl(0x0, 0x7FFEE177ABFC, 0x4) = -1 Err#1

42321/0xe91bc: stat64("/Users/sindenis/CSinDenis/university/sem3/OperatingSystems/lab05/c/./rt\_main\0", 0x7FFEE177B580, 0x0) = 0 0

42321/0xe91bc: getpid(0x0, 0x0, 0x0) = 42321 0

42321/0xe91bc: proc\_info(0x2, 0xA551, 0x16) = 1272 0

42321/0xe91bc: stat64("/usr/lib/libSystem.B.dylib\0", 0x7FFEE177A280, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libcache.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libcommonCrypto.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libcompiler\_rt.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libcopyfile.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libcorecrypto.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libdispatch.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libdyld.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libkeymgr.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/liblaunch.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libmacho.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libquarantine.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libremovefile.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_asl.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_blocks.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_c.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_configuration.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_coreservices.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_darwin.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_dnssd.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_info.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_m.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_malloc.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_networkextension.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_notify.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_sandbox.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_secinit.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_kernel.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_platform.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_pthread.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_symptoms.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libsystem\_trace.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libunwind.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/system/libxpc.dylib\0", 0x7FFEE1779EB0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/libobjc.A.dylib\0", 0x7FFEE17791B0, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/libc++abi.dylib\0", 0x7FFEE1778890, 0x0) = 0 0

42321/0xe91bc: stat64("/usr/lib/libc++.1.dylib\0", 0x7FFEE1778890, 0x0) = 0 0

42321/0xe91bc: open("/dev/dtracehelper\0", 0x2, 0xFFFFFFFFE177B010) = 3 0

42321/0xe91bc: ioctl(0x3, 0x80086804, 0x7FFEE177AE20) = 0 0

42321/0xe91bc: close(0x3) = 0 0

42321/0xe91bc: access("/AppleInternal/XBS/.isChrooted\0", 0x0, 0x0) = -1 Err#2

42321/0xe91bc: bsdthread\_register(0x7FFF60845400, 0x7FFF608453F0, 0x2000) = 1073742047 0

42321/0xe91bc: sysctlbyname(kern.bootargs, 0xD, 0x7FFEE177A270, 0x7FFEE177A260, 0x0) = 0 0

42321/0xe91bc: issetugid(0x0, 0x0, 0x0) = 0 0

42321/0xe91bc: ioctl(0x2, 0x4004667A, 0x7FFEE1779A74) = 0 0

42321/0xe91bc: mprotect(0x10E48B000, 0x1000, 0x0) = 0 0

42321/0xe91bc: mprotect(0x10E492000, 0x1000, 0x0) = 0 0

42321/0xe91bc: mprotect(0x10E493000, 0x1000, 0x0) = 0 0

42321/0xe91bc: mprotect(0x10E49A000, 0x1000, 0x0) = 0 0

42321/0xe91bc: mprotect(0x10E489000, 0x90, 0x1) = 0 0

42321/0xe91bc: mprotect(0x10E49B000, 0x1000, 0x1) = 0 0

42321/0xe91bc: mprotect(0x10E489000, 0x90, 0x3) = 0 0

42321/0xe91bc: mprotect(0x10E489000, 0x90, 0x1) = 0 0

42321/0xe91bc: getentropy(0x7FFEE1779BC0, 0x20, 0x0) = 0 0

42321/0xe91bc: getpid(0x0, 0x0, 0x0) = 42321 0

42321/0xe91bc: stat64("/AppleInternal\0", 0x7FFEE177A6E0, 0x0) = -1 Err#2

42321/0xe91bc: csops(0xA551, 0x7, 0x7FFEE177A210) = -1 Err#22

42321/0xe91bc: proc\_info(0x2, 0xA551, 0xD) = 64 0

42321/0xe91bc: csops(0xA551, 0x7, 0x7FFEE1779A50) = -1 Err#22

42321/0xe91bc: stat64("libstr\_array.so\0", 0x7FFEE177A540, 0x0) = 0 0

42321/0xe91bc: stat64("libstr\_array.so\0", 0x7FFEE177A980, 0x0) = 0 0

42321/0xe91bc: open("libstr\_array.so\0", 0x0, 0x0) = 3 0

42321/0xe91bc: fcntl(0x3, 0x62, 0x7FFEE1772110) = 0 0

42321/0xe91bc: mmap(0x10E49C000, 0x1000, 0x5, 0x12, 0x3, 0x0) = 0x10E49C000 0

42321/0xe91bc: mmap(0x10E49D000, 0x1000, 0x3, 0x12, 0x3, 0x1000) = 0x10E49D000 0

42321/0xe91bc: mmap(0x10E49E000, 0x218, 0x1, 0x12, 0x3, 0x2000) = 0x10E49E000 0

42321/0xe91bc: fcntl(0x3, 0x32, 0x7FFEE17723A0) = 0 0

42321/0xe91bc: close(0x3) = 0 0

42321/0xe91bc: getrlimit(0x1008, 0x7FFEE177B6A0, 0x0) = 0 0

42321/0xe91bc: fstat64(0x1, 0x7FFEE177B688, 0x0) = 0 0