

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Институт №8 “Компьютерные науки и прикладная математика”
Кафедра №806 “Вычислительная математика и программирование”

Лабораторная работа №3 по курсу
«Операционные системы»

Группа: М8О-210Б-23

Студент: Мишин С.А.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 17.12.24

Москва, 2024

Постановка задачи

Вариант 18.

Реализовать функционал из первой лабораторной используя shared memory и memory mapping. Для синхронизации чтения и записи использовать семафоры.

Родительский процесс создает два дочерних процесса. Первой строкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для открытия File с таким именем на запись для child1. Аналогично для второй строки и процесса child2. Родительский и дочерний процесс должны быть представлены разными программами.

Родительский процесс принимает от пользователя строки произвольной длины и пересылает их в pipe1 или в pipe2 в зависимости от правила фильтрации. Процесс child1 и child2 производят работу над строками. Процессы пишут результаты своей работы в стандартный вывод.

Правило фильтрации: нечетные строки отправляются в pipe1, четные в pipe2. Дочерние процессы удаляют все гласные из строк.

Общий метод и алгоритм решения

Использованные системные вызовы:

- `int write(int fildes, const void *buf, size_t nbyte);` – записывает данные из буфера в файловый дескриптор.
- `int read(int fildes, void *buf, size_t nbyte);` – читает данные из файлового дескриптора в буфер.
- `int close(int fildes);` – закрывает дескриптор.
- `int shm_open(const char *name, int oflag, ...);` – открывает объект shared memory.
- `int shm_unlink(const char *name);` – удаляет объект shared memory.
- `void* mmap(void *addr, size_t len, int prot, int flags, int fd, off_t offset);` – выделяет память, размещает файл или устройство в память процесса.
- `void* munmap(void *addr, size_t len);` – удаляет размещение.
- `int ftruncate(int fildes, off_t length);` – задаёт размер файла.
- `sem_t* sem_open(const char *name, int oflag, ...);` – создаёт и открывает именованный семафор.
- `sem_t* sem_close(sem_t *sem);` – закрывает именованный семафор.
- `int sem_unlink(const char *name);` – удаляет именованный семафор.
- `pid_t fork(void);` – создаёт новый процесс.
- `int execl(const char *path, const char *arg0, ..., /*, (char *)0, */);` – передаёт управление в другой исполняемый файл.

Программа создаёт объект shared memory для записи входной строки и чтения изменённой. Далее открывается два файла, создаётся четыре семафора (два на каждый процесс, один для синхронизации записи «P2C», другой для чтения, «C2P») и два процесса. Все семафоры изначально заблокированы.

Дочерний процесс ожидает семафор P2C, читает null-терминированную строку из shared memory, записывает результат и разблокирует семафор C2P. Родительский процесс читает строку из стандартного ввода, записывает её в shared memory и разблокирует P2C, ожидает C2P, читает изменённую строку из shared memory и записывает в файл. По окончании ввода shared memory обнуляется, все семафоры разблокируются и дочерние процессы завершаются.

Таким образом реализован аналог блокирующего чтения и записи в каналы посредством shared memory и двух семафоров.

Код программы

server/main.cpp

```
#include <algorithm>
#include <format>
#include <string>
#include <vector>

// Shared Memory
#include <fcntl.h>
#include <sys/mman.h>

// Semaphores
#include <semaphore.h>

#include "lab.h"
#include "posix_buf.h"

static const size_t CHILDREN = 2;

class SharedMemory {
    /** Path to the shared memory file. */
    std::string path_;
    /** File descriptor of the shared memory object. */
    int fd_;
    /** Mapped buffer. */
    char* buf_;
    /** Size of the mapped buffer. */
    size_t size_;

public:
    SharedMemory() : fd_(-1), buf_(nullptr), size_(0) {}

    // copying this shouldn't be possible
    SharedMemory(const SharedMemory&) = delete;
    SharedMemory& operator=(const SharedMemory&) = delete;

    explicit SharedMemory(const std::string& path, size_t size)
        : path_(path), size_(size) {
        fd_ = shm_open(path.c_str(), O_CREAT | O_RDWR, S_IRUSR | S_IWUSR);
        if (fd_ == -1) {
            throw std::runtime_error("can't create shared memory object");
        }

        if (ftruncate(fd_, static_cast<off_t>(size)) == -1) {
            throw std::runtime_error("can't resize shared memory");
        }
    }
}
```

```

        buf_ = static_cast<char*>(
            mmap(nullptr, size, PROT_READ | PROT_WRITE, MAP_SHARED, fd_, 0));
        if (buf_ == MAP_FAILED) {
            throw std::runtime_error("can't map shared memory");
        }
    }

~SharedMemory() {
    munmap(buf_, size_);
    shm_unlink(path_.c_str());
}

const std::string& path() const { return path_; }

char* buf() { return buf_; }

size_t size() const { return size_; }
};

struct Child {
    /** Child process ID. */
    pid_t pid_;
    /** Output file descriptor. */
    int output_fd_;
    /** Path to the P2C semaphore. */
    std::string lock_path_p2c_;
    /** Semaphore for parent-to-child communication. */
    sem_t* lock_p2c_;
    /** Path to the C2P semaphore. */
    std::string lock_path_c2p_;
    /** Semaphore for child-to-parent communication. */
    sem_t* lock_c2p_;

    Child() : pid_(-1), output_fd_(-1), lock_p2c_(nullptr), lock_c2p_(nullptr) {}

    ~Child() {
        close(output_fd_);
        sem_close(lock_p2c_);
        sem_unlink(lock_path_p2c_.c_str());
        sem_close(lock_c2p_);
        sem_unlink(lock_path_c2p_.c_str());
    }
};

int main(int argc, char* argv[]) {
    if (argc != 2) {
        perr << std::format("usage: {} <client program>", argv[0]);
        return 1;
    }
}

```

```

char* childPath = argv[1];

// Initialize and map shared memory.
SharedMemory shm(std::format("/line_buffer_{}", getpid()), MAX_LINE);

std::vector<Child> children(CHILDREN);

// Prompt the user for filenames and open them.
for (size_t i = 0; i != CHILDREN; ++i) {
    auto& child = children[i];

    pout << "Enter filename of file " << (i + 1) << ": " << std::endl;

    std::string filename;
    pin >> filename;

    int fd = open(filename.c_str(), O_CREAT | O_WRONLY | O_TRUNC, S_IRUSR | S_IWUSR);
    if (fd == -1) {
        perr << "can't open the file for writing" << std::endl;
        return 5;
    }

    child.output_fd_ = fd;
}

// Create semaphores.
for (size_t i = 0; i != CHILDREN; ++i) {
    auto& child = children[i];

    std::string lockPathP2c = std::format("/lock_{}_p2c", i);
    child.lock_path_p2c_ = lockPathP2c;

    sem_t* lockP2c = sem_open(lockPathP2c.c_str(), O_CREAT, S_IRUSR | S_IWUSR, 0);
    if (!lockP2c) {
        perr << "can't create P2C semaphore" << std::endl;
        return 6;
    }
    child.lock_p2c_ = lockP2c;

    std::string lockPathC2p = std::format("/lock_{}_c2p", i);
    child.lock_path_c2p_ = lockPathC2p;

    sem_t* lockC2p = sem_open(lockPathC2p.c_str(), O_CREAT, S_IRUSR | S_IWUSR, 0);
    if (!lockC2p) {
        perr << "can't create C2P semaphore" << std::endl;
        return 7;
    }
    child.lock_c2p_ = lockC2p;
}

```

```

// Start children.
for (auto& child : children) {
    pid_t pid = fork();
    if (pid == -1) {
        perr << "can't fork() to create a child process" << std::endl;
        return 7;
    }
    if (pid == 0) {
        if (execl(/* path */ childPath, /* argv[0] */ childPath,
                  /* argv[1] */ shm.path().c_str(),
                  /* argv[2] */ child.lock_path_p2c_.c_str(),
                  /* argv[3] */ child.lock_path_c2p_.c_str(), nullptr) == -1) {
            perr << "can't start child process" << std::endl;
            return 8;
        }
    } else {
        child.pid_ = pid;
    }
}

// Read lines from standard input and write them to shared memory. Notify children
// that the buffer changed. Wait for a response and write it to the output file.
size_t lines = 0;

std::string line;
while (std::getline(pin, line)) {
    auto& child = children[lines % children.size()];

    if (line.empty()) {
        continue;
    }

    if (line.length() > MAX_LINE) {
        perr << "Input line is too large. Please input a line containing less "
              "characters."
              << std::endl;
        continue;
    }

    // Copy the null-terminated string to shared memory.
    std::strncpy(shm.buf(), line.c_str(), shm.size());

    // Notify the child process that the data is ready.
    //
    pout << "[parent] notifying child" << std::endl;
    sem_post(child.lock_p2c_);

    // Wait for a response from the child process.
    //
    pout << "[parent] waiting for response" << std::endl;
    sem_wait(child.lock_c2p_);

    // Read response from shared memory and write it to the file.

```

```

        std::string response(shm.buf());
//        pout << "[parent] response: " << response << std::endl;

        response.push_back('\n');

        size_t written = write(child.output_fd_, response.data(), response.length());
        if (written != response.length()) {
            perr << "can't write to output file" << std::endl;
        }

        ++lines;
    }

    // EOF received -- zero the shared memory and notify all children to exit.
    std::fill(shm.buf(), shm.buf() + shm.size(), '\0');

    for (auto& child : children) {
        sem_post(child.lock_p2c_);
    }

    while (wait(nullptr) > 0)
        ;
}

```

client/main.cpp

```

#include <format>

// Shared Memory
#include <fcntl.h>
#include <sys/mman.h>

// Semaphore
#include <semaphore.h>

#include "lab.h"
#include "posix_buf.h"

int main(int argc, char* argv[]) {
    if (argc != 4) {
        perr << std::format(
            "usage: {} <shared memory path> <p2c semaphore path> <c2p semaphore path>\n",
            argv[0]);
        return 1;
    }

    const pid_t pid = getpid();

    sem_t* p2c = sem_open(argv[2], 0);
    if (!p2c) {
        perr << "can't open parent-to-child semaphore" << std::endl;
        return 2;
    }
}

```

```

}

sem_t* c2p = sem_open(argv[3], 0);
if (!c2p) {
    perr << "can't open child-to-parent semaphore" << std::endl;
    return 3;
}

int shm = shm_open(argv[1], O_RDWR, S_IRUSR | S_IWUSR);
if (shm == -1) {
    perr << "can't open shared memory" << std::endl;
    return 4;
}

char* shmBuf = static_cast<char*>(
    mmap(nullptr, MAX_LINE, PROT_READ | PROT_WRITE, MAP_SHARED, shm, 0));
if (shmBuf == MAP_FAILED) {
    perr << "can't map shared memory" << std::endl;
    return 5;
}

while (true) {
//    pout << "[pid: " << pid << "] waiting for request" << std::endl;

    // Wait for a signal from parent.
    sem_wait(p2c);

    // Exit signal.
    if (*shmBuf == '\0') {
        break;
    }

    std::string input(shmBuf);

    // Erase vowels from the input.
    std::erase_if(input, [](char c) {
        bool lowercase = c == 'a' || c == 'e' || c == 'o' || c == 'i' || c == 'u';
        bool uppercase = c == 'A' || c == 'E' || c == 'O' || c == 'I' || c == 'U';

        return lowercase || uppercase;
    });

    // Write back the result.
    std::strncpy(shmBuf, input.c_str(), MAX_LINE);

//    pout << "[pid: " << pid << "] notifying parent" << std::endl;

    // Notify parent of the result.
    sem_post(c2p);
}

```



```

        munmap(shmBuf, MAX_LINE);
        sem_close(p2c);
        sem_close(c2p);
    }

```

Протокол работы программы

Тестирование:

```

/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-build-release/src/lab-
3/server/lab_3_server ../client/lab_3_client

```

Enter filename of file 1:

test1.txt

Enter filename of file 2:

test2.txt

1 ABABABABABABBA

2 BABABABABABABA

3 JKASJDKSDJIWQEJ

4 kjsadkfjasflkja

5 ababababbaba

6 babababababba

7 ksadjfdksajfiqer

8 c

9 a

10 b

^D

skidunion@apollo server % cat test1.txt

1 BBBBBBBB

3 JKSDKSDJWQJ

5 bbbbbb

7 ksdjfdksjfqr

9

skidunion@apollo server % cat test2.txt

2 BBBBBBBB

4 kjsdkfjsflkj

6 bbbbbbbb

8 c

10 b

Dtrace:

| PID/THRD | RELATIVE | ELAPSD | CPU SYSCALL(args) | = return |
|--------------|----------|--------|--------------------------------|----------|
| 1660/0x2f04: | 341: | 0: | 0 fork() | = 0 0 |
| 1660/0x2f04: | 346 | 29 | 5 munmap(0x104594000, 0x8C000) | = 0 0 |
| 1660/0x2f04: | 351 | 5 | 4 munmap(0x104620000, 0x8000) | = 0 0 |

| | | | | |
|-----------------------------------|-------|----|---|-------------|
| 1660/0x2f04: | 353 | 2 | 1 munmap(0x104628000, 0x4000) | = 0 0 |
| 1660/0x2f04: | 355 | 2 | 2 munmap(0x10462C000, 0x4000) | = 0 0 |
| 1660/0x2f04: | 357 | 2 | 1 munmap(0x104630000, 0x50000) | = 0 0 |
| 1660/0x2f04: | 370 | 1 | 0 crossarch_trap(0x0, 0x0, 0x0) | = -1 Err#45 |
| 1660/0x2f04: | 396 | 25 | 24 open("./\0", 0x100000, 0x0) | = 3 0 |
| 1660/0x2f04: | 399 | 3 | 2 fcntl(0x3, 0x32, 0x16BA77278) | = 0 0 |
| 1660/0x2f04: | 400 | 1 | 1 close(0x3) | = 0 0 |
| 1660/0x2f04: | 407 | 6 | 5 fsgetpath(0x16BA77288, 0x400, 0x16BA77268) | = 100 0 |
| 1660/0x2f04: | 418 | 11 | 10 fsgetpath(0x16BA77298, 0x400, 0x16BA77278) | = 14 0 |
| 1660/0x2f04: | 423 | 1 | 0 csrctl(0x0, 0x16BA7769C, 0x4) | = -1 Err#1 |
| 0 0 1660/0x2f04: | 426 | 3 | 2 __mac_syscall(0x18ADCDACF, 0x2, 0x16BA775E0) | = |
| 1660/0x2f04: | 427 | 0 | 0 csrctl(0x0, 0x16BA7768C, 0x4) | = -1 Err#1 |
| 0 0 1660/0x2f04: | 431 | 2 | 2 __mac_syscall(0x18ADCA902, 0x5A, 0x16BA77620) | = |
| 1660/0x2f04: 0x18ADCC553, 0xD) | 444 | 7 | 6 sysctl([unknown, 3, 0, 0, 0, 0] (2), 0x16BA76BA8, 0x16BA76BA0, | |
| | = 0 0 | | | |
| 1660/0x2f04: 0x0, 0x0) | 445 | 1 | 1 sysctl([CTL_KERN, 140, 0, 0, 0, 0] (2), 0x16BA76C58, 0x16BA76C50, | |
| | = 0 0 | | | |
| 1660/0x2f04: | 451 | 4 | 3 open("/\0", 0x20100000, 0x0) | = 3 0 |
| 1660/0x2f04: = 4 0 | 466 | 15 | 14 openat(0x3, "System/Cryptexes/OS\0", 0x100000, 0x0) | |
| 1660/0x2f04: | 468 | 1 | 0 dup(0x4, 0x0, 0x0) | = 5 0 |
| 1660/0x2f04: | 471 | 3 | 2 fstatat64(0x4, 0x16BA76731, 0x16BA766A0) | = 0 0 |
| 1660/0x2f04: = 6 0 | 476 | 4 | 3 openat(0x4, "System/Library/dyld/\0", 0x100000, 0x0) | |
| 1660/0x2f04: | 477 | 1 | 0 fcntl(0x6, 0x32, 0x16BA76730) | = 0 0 |
| 1660/0x2f04: | 477 | 0 | 0 dup(0x6, 0x0, 0x0) | = 7 0 |
| 1660/0x2f04: | 478 | 0 | 0 dup(0x5, 0x0, 0x0) | = 8 0 |
| 1660/0x2f04: | 479 | 1 | 0 close(0x3) | = 0 0 |
| 1660/0x2f04: | 480 | 1 | 0 close(0x5) | = 0 0 |
| 1660/0x2f04: | 480 | 0 | 0 close(0x4) | = 0 0 |
| 1660/0x2f04: | 480 | 0 | 0 close(0x6) | = 0 0 |
| 0 0 1660/0x2f04: | 482 | 2 | 1 __mac_syscall(0x18ADCDACF, 0x2, 0x16BA77120) | = |
| 0 0 1660/0x2f04: | 484 | 2 | 1 shared_region_check_np(0x16BA76D40, 0x0, 0x0) | = |
| 1660/0x2f04: | 487 | 3 | 2 fsgetpath(0x16BA772A0, 0x400, 0x16BA771E8) | = 82 0 |
| 1660/0x2f04: | 488 | 0 | 0 fcntl(0x8, 0x32, 0x16BA772A0) | = 0 0 |
| 1660/0x2f04: | 489 | 0 | 0 close(0x8) | = 0 0 |
| 1660/0x2f04: | 490 | 0 | 0 close(0x7) | = 0 0 |
| 1660/0x2f04: | 496 | 3 | 2 getfsstat64(0x0, 0x0, 0x2) | = 10 0 |
| 1660/0x2f04: | 503 | 7 | 6 getfsstat64(0x1047B6110, 0x54B0, 0x2) | = 10 0 |
| 0 0 1660/0x2f04: | 509 | 6 | 5 getattrlist("/\0", 0x16BA771D0, 0x16BA77140) | = |

```

1660/0x2f04:          518          7          6
stat64("/System/Volumes/Preboot/Cryptexes/OS/System/Library/dyld/dyld_shared_cache_arm64e\0",
0x16BA77530, 0x0)          = 0 0

dtrace: error on enabled probe ID 1690 (ID 845: syscall::stat64:return): invalid address (0x0) in
action #12 at DIF offset 12

1660/0x2f04:      554      3      2 stat64("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/server/lab_3_server\0", 0x16BA769E0, 0x0)          = 0 0

1660/0x2f04:      567      13      12 open("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/server/lab_3_server\0", 0x0, 0x0)          = 3 0

1660/0x2f04:      576      32      8 mmap(0x0, 0x1F9B0, 0x1, 0x40002, 0x3, 0x0)          = 0x1047F8000
0

1660/0x2f04:      578      1      0 fcntl(0x3, 0x32, 0x16BA76AF8)          = 0 0

1660/0x2f04:      578      1      0 close(0x3)          = 0 0

1660/0x2f04:      581      2      1 munmap(0x1047F8000, 0x1F9B0)          = 0 0

1660/0x2f04:      583      2      1 stat64("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/server/lab_3_server\0", 0x16BA76F50, 0x0)          = 0 0

1660/0x2f04:      603      3      2 stat64("/usr/lib/libc++.1.dylib\0", 0x16BA75E90, 0x0)
= -1 Err#2

1660/0x2f04:          608          2          2
stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/libc++.1.dylib\0", 0x16BA75E40, 0x0)
-1 Err#2          =

1660/0x2f04:      650      2      1 stat64("/usr/lib/system/libdispatch.dylib\0", 0x16BA73AB0, 0x0)
= -1 Err#2

1660/0x2f04:          651          1          1
stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/system/libdispatch.dylib\0", 0x16BA73A60, 0x0)
= -1 Err#2

1660/0x2f04:      652      0      0 stat64("/usr/lib/system/libdispatch.dylib\0", 0x16BA73AB0, 0x0)
= -1 Err#2

1660/0x2f04:      661      1      0 stat64("/usr/lib/libSystem.B.dylib\0", 0x16BA75E90, 0x0)
= -1 Err#2

1660/0x2f04:          665          1          0
stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/libSystem.B.dylib\0", 0x16BA75E40, 0x0)
-1 Err#2          =

Enter filename of file 1:

1660/0x2f04:      780      45      21 open("/dev/dtracehelper\0", 0x2, 0x0)          = 3 0

1660/0x2f04:      1246      1808      466 ioctl(0x3, 0x80086804, 0x16BA75A88)          = 0 0

1660/0x2f04:      1249      3      2 close(0x3)          = 0 0

1660/0x2f04:      1472      12      11 open("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/server/lab_3_server\0", 0x0, 0x0)          = 3 0

1660/0x2f04:      1475      3      2 __mac_syscall(0x18ADCDAF, 0x2, 0x16BA75190)          =
0 0

1660/0x2f04:      1487      11      9 map_with_linking_np(0x16BA74CE0, 0x1, 0x16BA74D10)          =
0 0

1660/0x2f04:      1487      1      0 close(0x3)          = 0 0

1660/0x2f04:      1489      2      1 mprotect(0x10439C000, 0x4000, 0x1)          = 0 0

1660/0x2f04:      1510      1      0 shared_region_check_np(0xFFFFFFFFFFFFFFFF, 0x0, 0x0)
= 0 0

1660/0x2f04:      1512      1      1 mprotect(0x1047B4000, 0x40000, 0x1)          = 0 0

1660/0x2f04:      1523      2      1 access("/AppleInternal/XBS/.isChrooted\0", 0x0, 0x0)
= -1 Err#2

1660/0x2f04:      1536      3      1 bsdthread_register(0x18B0D0D2C, 0x18B0D0D20, 0x4000)
= 1073746399 0

1660/0x2f04:      1561      1      0 getpid(0x0, 0x0, 0x0)          = 1660 0

```

| | | | | | |
|-----|----------------------------|------|----|--|---------------|
| 3 0 | 1660/0x2f04: | 1565 | 4 | 2 shm_open(0x18AF6BF51, 0x0, 0xFFFFFFFF8AF721E4) | = |
| | 1660/0x2f04: | 1566 | 1 | 0 fstat64(0x3, 0x16BA75E20, 0x0) | = 0 0 |
| 0 | 1660/0x2f04: | 1569 | 4 | 2 mmap(0x0, 0x4000, 0x1, 0x40001, 0x3, 0x0) | = 0x104800000 |
| | 1660/0x2f04: | 1571 | 1 | 0 close(0x3) | = 0 0 |
| | 1660/0x2f04: | 1582 | 1 | 0 ioctl(0x2, 0x4004667A, 0x16BA75ECC) | = 0 0 |
| | 1660/0x2f04: | 1588 | 1 | 0 mprotect(0x10480C000, 0x4000, 0x0) | = 0 0 |
| | 1660/0x2f04: | 1589 | 1 | 0 mprotect(0x104818000, 0x4000, 0x0) | = 0 0 |
| | 1660/0x2f04: | 1593 | 1 | 0 mprotect(0x10481C000, 0x4000, 0x0) | = 0 0 |
| | 1660/0x2f04: | 1594 | 1 | 0 mprotect(0x104828000, 0x4000, 0x0) | = 0 0 |
| | 1660/0x2f04: | 1597 | 1 | 0 mprotect(0x10482C000, 0x4000, 0x0) | = 0 0 |
| | 1660/0x2f04: | 1598 | 1 | 0 mprotect(0x104838000, 0x4000, 0x0) | = 0 0 |
| | 1660/0x2f04: | 1603 | 1 | 0 mprotect(0x104804000, 0xA0, 0x1) | = 0 0 |
| | 1660/0x2f04: | 1604 | 1 | 1 mprotect(0x104804000, 0xA0, 0x3) | = 0 0 |
| | 1660/0x2f04: | 1607 | 1 | 0 mprotect(0x104804000, 0xA0, 0x1) | = 0 0 |
| | 1660/0x2f04: | 1610 | 1 | 0 mprotect(0x10483C000, 0x4000, 0x1) | = 0 0 |
| | 1660/0x2f04: | 1614 | 1 | 0 mprotect(0x104840000, 0xA0, 0x1) | = 0 0 |
| | 1660/0x2f04: | 1618 | 1 | 0 mprotect(0x104840000, 0xA0, 0x3) | = 0 0 |
| | 1660/0x2f04: | 1620 | 1 | 0 mprotect(0x104840000, 0xA0, 0x1) | = 0 0 |
| | 1660/0x2f04: | 1621 | 1 | 0 mprotect(0x104804000, 0xA0, 0x3) | = 0 0 |
| | 1660/0x2f04: | 1624 | 1 | 0 mprotect(0x104804000, 0xA0, 0x1) | = 0 0 |
| | 1660/0x2f04: | 1625 | 1 | 0 mprotect(0x10483C000, 0x4000, 0x3) | = 0 0 |
| | 1660/0x2f04: | 1627 | 1 | 0 mprotect(0x10483C000, 0x4000, 0x1) | = 0 0 |
| | 1660/0x2f04: | 1628 | 1 | 0 mprotect(0x1047B4000, 0x40000, 0x3) | = 0 0 |
| | 1660/0x2f04: | 1631 | 1 | 0 mprotect(0x1047B4000, 0x40000, 0x1) | = 0 0 |
| | 1660/0x2f04: = -1 Err#5 | 1638 | 1 | 0 objc_bp_assist_cfg_np(0x18ACFD000, 0x80000018001C1048, 0x0) | |
| | 1660/0x2f04: | 1640 | 1 | 0 issetugid(0x0, 0x0, 0x0) | = 0 0 |
| | 1660/0x2f04: | 1655 | 1 | 0 mprotect(0x1047B4000, 0x40000, 0x3) | = 0 0 |
| | 1660/0x2f04: | 1695 | 11 | 10 getentropy(0x16BA75538, 0x20, 0x0) | = 0 0 |
| | 1660/0x2f04: | 1723 | 1 | 0 mprotect(0x1047B4000, 0x40000, 0x1) | = 0 0 |
| | 1660/0x2f04: | 1738 | 1 | 0 mprotect(0x1047B4000, 0x40000, 0x3) | = 0 0 |
| | 1660/0x2f04: | 1740 | 1 | 0 mprotect(0x1047B4000, 0x40000, 0x1) | = 0 0 |
| | 1660/0x2f04: | 1748 | 6 | 4 getattrlist("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-build-release/src/lab-3/server/lab_3_server\0", 0x16BA75DB0, 0x16BA75DC8) | = 0 0 |
| | 1660/0x2f04: | 1756 | 7 | 6 access("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-build-release/src/lab-3/server\0", 0x4, 0x0) | = 0 0 |
| | 1660/0x2f04: | 1763 | 7 | 7 open("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-build-release/src/lab-3/server\0", 0x0, 0x0) | = 3 0 |
| | 1660/0x2f04: | 1764 | 1 | 0 fstat64(0x3, 0x15AE04450, 0x0) | = 0 0 |
| | 1660/0x2f04: | 1765 | 0 | 0 csrctl(0x0, 0x16BA75FDC, 0x4) | = 0 0 |
| | 1660/0x2f04: | 1767 | 2 | 1 fcntl(0x3, 0x32, 0x16BA75C98) | = 0 0 |
| | 1660/0x2f04: | 1768 | 1 | 0 close(0x3) | = 0 0 |

```

1660/0x2f04:      1772      2      1 open("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/server/Info.plist\0", 0x0, 0x0)      = -1 Err#2

1660/0x2f04:      1782      1      0 proc_info(0x2, 0x67C, 0xD)      = 64 0

1660/0x2f04:      1795      3      1 csops_audittoken(0x67C, 0x10, 0x16BA76020)      = 0 0

1660/0x2f04:      1805      5      4 sysctl([unknown, 3, 0, 0, 0, 0] (2), 0x16BA76378, 0x16BA76370,
0x18E502D3D, 0x15) = 0 0

1660/0x2f04:      1806      2      1 sysctl([CTL_KERN, 138, 0, 0, 0, 0] (2), 0x16BA76408, 0x16BA76400,
0x0, 0x0)      = 0 0

1660/0x2f04:      1808      1      0 csops(0x67C, 0x0, 0x16BA764AC)      = 0 0

1660/0x2f04:      1810      1      0 mprotect(0x1047B4000, 0x40000, 0x3)      = 0 0

1660/0x2f04:      1871      27      3 shm_open(0x16BA77670, 0x202, 0x180)      = 3 0

1660/0x2f04:      1874      3      1 ftruncate(0x3, 0x400, 0x0)      = 0 0

1660/0x2f04:      1875      2      1 mmap(0x0, 0x400, 0x3, 0x40001, 0x3, 0x0)      = 0x1043B0000

0

1660/0x2f04:      1883      2      1 write(0x1, "Enter filename of file 1: \n\0", 0x1B)      =
27 0

test1.txt

Enter filename of file 2:

1660/0x2f04:      1894 2524886      10 read(0x0, "test1.txt\n\0", 0x1000)      = 10 0

1660/0x2f04:      2142      479      235 open("test1.txt\0", 0x601, 0x180)      = 4 0

1660/0x2f04:      2172      67      8 write(0x1, "Enter filename of file 2: \n\0", 0x1B)      =
27 0

test2.txt

1660/0x2f04:      2186 2504944      12 read(0x0, "test2.txt\n\0", 0x1000)      = 10 0

1660/0x2f04:      2462      801      270 open("test2.txt\0", 0x601, 0x180)      = 5 0

1660/0x2f04:      2484      72      11 sem_open(0x16BA77658, 0x200, 0x180)      = 6 0

1660/0x2f04:      2496      5      3 sem_open(0x16BA77640, 0x200, 0x180)      = 7 0

1660/0x2f04:      2499      4      2 sem_open(0x16BA77658, 0x200, 0x180)      = 8 0

1660/0x2f04:      2504      5      4 sem_open(0x16BA77640, 0x200, 0x180)      = 9 0

1660/0x2f04:      3229      1684      680 fork()      = 1727 0

1727/0x31f4:      86:      0:      0 fork()      = 0 0

1727/0x31f4:      87      59      0 thread_selfid(0x0, 0x0, 0x0)      = 12788 0

1727/0x31f4:      89      3      0 bsdthread_register(0x18B0D0D2C, 0x18B0D0D20, 0x4000)
= -1 Err#22

1727/0x31f4:      97      6      4 mprotect(0x104840000, 0xA0, 0x3)      = 0 0

1727/0x31f4:      104      3      1 mprotect(0x104840000, 0xA0, 0x1)      = 0 0

1660/0x2f04:      3977      1615      607 fork()      = 1728 0

1728/0x31f6:      59:      0:      0 fork()      = 0 0

1728/0x31f6:      59      3      0 thread_selfid(0x0, 0x0, 0x0)      = 12790 0

1728/0x31f6:      61      4      1 bsdthread_register(0x18B0D0D2C, 0x18B0D0D20, 0x4000)
= -1 Err#22

1728/0x31f6:      68      6      4 mprotect(0x104840000, 0xA0, 0x3)      = 0 0

1728/0x31f6:      75      3      1 mprotect(0x104840000, 0xA0, 0x1)      = 0 0

```

dtrace: error on enabled probe ID 1688 (ID 287: syscall::execve:return): invalid address (0x16ba77af7) in action #12 at DIF offset 12

dtrace: error on enabled probe ID 1688 (ID 287: syscall::execve:return): invalid address (0x16ba77af7) in action #12 at DIF offset 12

| | | | | |
|--------------|--------------|------|--|-------------|
| 1728/0x31f8: | 698: | 0: | 0 fork() | = 0 0 |
| 1728/0x31f8: | 700 | 6 | 2 mprotect(0x103140000, 0x8000, 0x1) | = 0 0 |
| 1728/0x31f8: | 704 | 2 | 0 thread_selfid(0x0, 0x0, 0x0) | = 12792 0 |
| 1727/0x31f7: | 918: | 0: | 0 fork() | = 0 0 |
| 1728/0x31f8: | 728 | 2 | 0 crossarch_trap(0x0, 0x0, 0x0) | = -1 Err#45 |
| 1727/0x31f7: | 919 | 4 | 1 mprotect(0x1048BC000, 0x8000, 0x1) | = 0 0 |
| 1727/0x31f7: | 922 | 2 | 0 thread_selfid(0x0, 0x0, 0x0) | = 12791 0 |
| 1727/0x31f7: | 929 | 2 | 0 crossarch_trap(0x0, 0x0, 0x0) | = -1 Err#45 |
| 0 0 | 1728/0x31f8: | 791 | 63 60 shared_region_check_np(0x16CFFB780, 0x0, 0x0) | = |
| | 1728/0x31f8: | 809 | 2 0 thread_selfid(0x0, 0x0, 0x0) | = 12792 0 |
| | 1728/0x31f8: | 822 | 2 0 getpid(0x0, 0x0, 0x0) | = 1728 0 |
| | 1728/0x31f8: | 823 | 3 0 proc_info(0xF, 0x6C0, 0x0) | = 0 0 |
| | 1728/0x31f8: | 834 | 6 4 munmap(0x1030B4000, 0x8C000) | = 0 0 |
| | 1728/0x31f8: | 837 | 4 2 munmap(0x103140000, 0x8000) | = 0 0 |
| | 1728/0x31f8: | 840 | 3 1 munmap(0x103148000, 0x4000) | = 0 0 |
| | 1728/0x31f8: | 843 | 3 1 munmap(0x10314C000, 0x4000) | = 0 0 |
| | 1728/0x31f8: | 845 | 2 1 munmap(0x103150000, 0x50000) | = 0 0 |
| | 1728/0x31f8: | 874 | 2 0 crossarch_trap(0x0, 0x0, 0x0) | = -1 Err#45 |
| | 1728/0x31f8: | 926 | 51 49 open("./\0", 0x100000, 0x0) | = 3 0 |
| | 1728/0x31f8: | 929 | 4 2 fcntl(0x3, 0x32, 0x16CFFB248) | = 0 0 |
| | 1728/0x31f8: | 932 | 4 2 close(0x3) | = 0 0 |
| | 1728/0x31f8: | 942 | 9 7 fsgetpath(0x16CFFB258, 0x400, 0x16CFFB238) | = 100 0 |
| | 1728/0x31f8: | 958 | 16 14 fsgetpath(0x16CFFB268, 0x400, 0x16CFFB248) | = 14 0 |
| | 1728/0x31f8: | 966 | 2 0 csrctl(0x0, 0x16CFFB66C, 0x4) | = -1 Err#1 |
| 0 0 | 1728/0x31f8: | 970 | 5 3 __mac_syscall(0x18ADCDACF, 0x2, 0x16CFFB5B0) | = |
| | 1728/0x31f8: | 972 | 1 0 csrctl(0x0, 0x16CFFB65C, 0x4) | = -1 Err#1 |
| 0 0 | 1728/0x31f8: | 979 | 6 4 __mac_syscall(0x18ADCA902, 0x5A, 0x16CFFB5F0) | = |
| | 1728/0x31f8: | 1002 | 14 10 sysctl([unknown, 3, 0, 0, 0, 0] (2), 0x16CFFAB78, 0x16CFFAB70, 0x18ADCC553, 0xD) | = 0 0 |
| | 1728/0x31f8: | 1006 | 6 3 sysctl([CTL_KERN, 140, 0, 0, 0, 0] (2), 0x16CFFAC28, 0x16CFFAC20, 0x0, 0x0) | = 0 0 |
| | 1728/0x31f8: | 1021 | 12 10 open("/\0", 0x20100000, 0x0) | = 3 0 |
| 0 0 | 1727/0x31f7: | 993 | 481 62 shared_region_check_np(0x16B9B3780, 0x0, 0x0) | = |
| | 1728/0x31f8: | 1048 | 28 23 openat(0x3, "System/Cryptexes/OS\0", 0x100000, 0x0) | = 10 0 |
| | 1728/0x31f8: | 1050 | 3 1 dup(0xA, 0x0, 0x0) | = 11 0 |
| | 1728/0x31f8: | 1058 | 8 6 fstatat64(0xA, 0x16CFFA701, 0x16CFFA670) | = 0 0 |
| | 1728/0x31f8: | 1069 | 12 9 openat(0xA, "System/Library/dyld/\0", 0x100000, 0x0) | = 12 0 |

```

1728/0x31f8:      1072      3      2 fcntl(0xC, 0x32, 0x16CFFA700)          = 0 0
1727/0x31f7:      1011     58      0 thread_selfid(0x0, 0x0, 0x0)          = 12791 0
1728/0x31f8:      1073      1      0 dup(0xC, 0x0, 0x0)              = 13 0
1728/0x31f8:      1074      1      0 dup(0xB, 0x0, 0x0)              = 14 0
1728/0x31f8:      1076      2      1 close(0x3)                      = 0 0
1727/0x31f7:      1024      2      0 getpid(0x0, 0x0, 0x0)           = 1727 0
1728/0x31f8:      1077      1      0 close(0xB)                      = 0 0
1728/0x31f8:      1078      2      0 close(0xA)                      = 0 0
1727/0x31f7:      1025      2      0 proc_info(0xF, 0x6BF, 0x0)       = 0 0
1728/0x31f8:      1078      1      0 close(0xC)                      = 0 0
1727/0x31f7:      1034      6      4 munmap(0x104830000, 0x8C000)      = 0 0
0 0 1728/0x31f8:      1083      5      3 __mac_syscall(0x18ADCDA CF, 0x2, 0x16CFFB0F0)          =
0 0 1728/0x31f8:      1084      3      0 shared_region_check_np(0x16CFFAD10, 0x0, 0x0)          =
1727/0x31f7:      1039      5      4 munmap(0x1048BC000, 0x8000)       = 0 0
1727/0x31f7:      1041      3      1 munmap(0x1048C4000, 0x4000)       = 0 0
1728/0x31f8:      1092      6      5 fsgetpath(0x16CFFB270, 0x400, 0x16CFFB1B8)          = 82 0
1727/0x31f7:      1044      3      2 munmap(0x1048C8000, 0x4000)       = 0 0
1728/0x31f8:      1096      2      1 fcntl(0xE, 0x32, 0x16CFFB270)      = 0 0
1727/0x31f7:      1047      3      2 munmap(0x1048CC000, 0x50000)       = 0 0
1728/0x31f8:      1098      2      0 close(0xE)                      = 0 0
1728/0x31f8:      1099      1      0 close(0xD)                      = 0 0
1727/0x31f7:      1066      2      0 crossarch_trap(0x0, 0x0, 0x0)      = -1 Err#45
1728/0x31f8:      1113      6      4 getfsstat64(0x0, 0x0, 0x2)        = 10 0
1728/0x31f8:      1128     15     14 getfsstat64(0x103232110, 0x54B0, 0x2)          = 10 0
1727/0x31f7:      1095     28     27 open("./0", 0x100000, 0x0)        = 3 0
1727/0x31f7:      1098      3      2 fcntl(0x3, 0x32, 0x16B9B3248)      = 0 0
0 0 1728/0x31f8:      1139     12      9 getatrlist("./0", 0x16CFFB1A0, 0x16CFFB110)          =
1727/0x31f7:      1100      3      1 close(0x3)                      = 0 0
1727/0x31f7:      1105      5      3 fsgetpath(0x16B9B3258, 0x400, 0x16B9B3238)          = 100 0
1728/0x31f8:      1161      16      14
stat64("/System/Volumes/Preboot/Cryptexes/OS/System/Library/dyld/dyld_shared_cache_arm64e\0",
0x16CFFB500, 0x0)          = 0 0
1727/0x31f7:      1116     10      9 fsgetpath(0x16B9B3268, 0x400, 0x16B9B3248)          = 14 0
1727/0x31f7:      1122      2      0 csrctl(0x0, 0x16B9B366C, 0x4)      = -1 Err#1
dtrace: error on enabled probe ID 1690 (ID 845: syscall::stat64:return): invalid address (0x0) in
action #12 at DIF offset 12
0 0 1727/0x31f7:      1126      5      3 __mac_syscall(0x18ADCDA CF, 0x2, 0x16B9B35B0)          =
1727/0x31f7:      1127      1      0 csrctl(0x0, 0x16B9B365C, 0x4)      = -1 Err#1
0 0 1727/0x31f7:      1132      4      2 __mac_syscall(0x18ADCA902, 0x5A, 0x16B9B35F0)          =

```

```

1728/0x31f8:      1205      10      8 stat64("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client/lab_3_client\0", 0x16CFFA9B0, 0x0)      = 0 0

1727/0x31f7:      1146       7      4 sysctl([unknown, 3, 0, 0, 0, 0] (2), 0x16B9B2B78, 0x16B9B2B70,
0x18ADCC553, 0xD)      = 0 0

1727/0x31f7:      1149       4      2 sysctl([CTL_KERN, 140, 0, 0, 0, 0] (2), 0x16B9B2C28, 0x16B9B2C20,
0x0, 0x0)      = 0 0

1728/0x31f8:      1237      32     31 open("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client/lab_3_client\0", 0x0, 0x0)      = 3 0

1727/0x31f7:      1160       9      8 open("/\0", 0x20100000, 0x0)      = 3 0

1728/0x31f8:      1253      19     15 mmap(0x0, 0x1F400, 0x1, 0x40002, 0x3, 0x0)      = 0x103274000
0

1728/0x31f8:      1256       3      2 fcntl(0x3, 0x32, 0x16CFFAAC8)      = 0 0

1727/0x31f7:      1177      18     16 openat(0x3, "System/Cryptexes/OS\0", 0x100000, 0x0)
= 10 0

1728/0x31f8:      1257       2      1 close(0x3)      = 0 0

1727/0x31f7:      1178       2      0 dup(0xA, 0x0, 0x0)      = 11 0

1728/0x31f8:      1263       3      2 munmap(0x103274000, 0x1F400)      = 0 0

1727/0x31f7:      1184       6      4 fstatat64(0xA, 0x16B9B2701, 0x16B9B2670)      = 0 0

1728/0x31f8:      1268       6      4 stat64("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client/lab_3_client\0", 0x16CFFAF20, 0x0)      = 0 0

1727/0x31f7:      1194      11      9 openat(0xA, "System/Library/dyld/\0", 0x100000, 0x0)
= 12 0

1727/0x31f7:      1196       3      1 fcntl(0xC, 0x32, 0x16B9B2700)      = 0 0

1727/0x31f7:      1197       1      0 dup(0xC, 0x0, 0x0)      = 13 0

1727/0x31f7:      1198       1      0 dup(0xB, 0x0, 0x0)      = 14 0

1727/0x31f7:      1200       2      1 close(0x3)      = 0 0

1728/0x31f8:      1309       6      4 stat64("/usr/lib/libc++.1.dylib\0", 0x16CFF9E60, 0x0)
= -1 Err#2

1727/0x31f7:      1201       1      0 close(0xB)      = 0 0

1727/0x31f7:      1202       1      0 close(0xA)      = 0 0

1727/0x31f7:      1202       1      0 close(0xC)      = 0 0

1728/0x31f8:      1319       5      5 stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/libc++.1.dylib\0", 0x16CFF9E10, 0x0)
= -1 Err#2      3

1727/0x31f7:      1206       4      2 __mac_syscall(0x18ADCDACF, 0x2, 0x16B9B30F0)      =
0 0

1727/0x31f7:      1207       3      0 shared_region_check_np(0x16B9B2D10, 0x0, 0x0)      =
0 0

1727/0x31f7:      1211       4      2 fsgetpath(0x16B9B3270, 0x400, 0x16B9B31B8)      = 82 0

1727/0x31f7:      1214       2      0 fcntl(0xE, 0x32, 0x16B9B3270)      = 0 0

1727/0x31f7:      1215       1      0 close(0xE)      = 0 0

1727/0x31f7:      1216       1      0 close(0xD)      = 0 0

1727/0x31f7:      1224       4      2 getfsstat64(0x0, 0x0, 0x2)      = 10 0

1727/0x31f7:      1243      19     17 getfsstat64(0x10487A110, 0x54B0, 0x2)      = 10 0

1727/0x31f7:      1250       7      5 getatrrlist("/\0", 0x16B9B31A0, 0x16B9B3110)      =
0 0

1727/0x31f7:      1264      10     10 stat64("/System/Volumes/Preboot/Cryptexes/OS/System/Library/dyld/dyld_shared_cache_arm64e\0",
0x16B9B3500, 0x0)      = 0 0      8

```



```

1728/0x31f8:      1438      4      3 stat64("/usr/lib/system/libdispatch.dylib\0", 0x16CFF7A80, 0x0)
= -1 Err#2

dtrace: error on enabled probe ID 1690 (ID 845: syscall::stat64:return): invalid address (0x0) in
action #12 at DIF offset 12

1728/0x31f8:      1442      4      2
stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/system/libdispatch.dylib\0", 0x16CFF7A30, 0x0)
= -1 Err#2

1728/0x31f8:      1444      2      1 stat64("/usr/lib/system/libdispatch.dylib\0", 0x16CFF7A80, 0x0)
= -1 Err#2

1727/0x31f7:      1291      6      5 stat64("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client/lab_3_client\0", 0x16B9B29B0, 0x0) = 0 0

1728/0x31f8:      1465      3      2 stat64("/usr/lib/libSystem.B.dylib\0", 0x16CFF9E60, 0x0)
= -1 Err#2

1727/0x31f7:      1312      21     20 open("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client/lab_3_client\0", 0x0, 0x0) = 3 0

1727/0x31f7:      1320      8      5 mmap(0x0, 0x1F400, 0x1, 0x40002, 0x3, 0x0) = 0x1048BC000
0

1727/0x31f7:      1323      3      1 fcntl(0x3, 0x32, 0x16B9B2AC8) = 0 0

1727/0x31f7:      1324      2      0 close(0x3) = 0 0

1727/0x31f7:      1329      3      2 munmap(0x1048BC000, 0x1F400) = 0 0

1727/0x31f7:      1335      6      4 stat64("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client/lab_3_client\0", 0x16B9B2F20, 0x0) = 0 0

1727/0x31f7:      1363      4      2 stat64("/usr/lib/libc++.1.dylib\0", 0x16B9B1E60, 0x0)
= -1 Err#2

1727/0x31f7:      1372      4      2
stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/libc++.1.dylib\0", 0x16B9B1E10, 0x0)
= -1 Err#2

1727/0x31f7:      1411      3      2 stat64("/usr/lib/system/libdispatch.dylib\0", 0x16B9AFA80, 0x0)
= -1 Err#2

1727/0x31f7:      1414      3      2
stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/system/libdispatch.dylib\0", 0x16B9AFA30, 0x0)
= -1 Err#2

1727/0x31f7:      1426      4      2 stat64("/usr/lib/system/libdispatch.dylib\0", 0x16B9AFA80, 0x0)
= -1 Err#2

1727/0x31f7:      1445      3      1 stat64("/usr/lib/libSystem.B.dylib\0", 0x16B9B1E60, 0x0)
= -1 Err#2

1727/0x31f7:      1455      3      2
stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/libSystem.B.dylib\0", 0x16B9B1E10, 0x0)
= -1 Err#2

1728/0x31f8:      1495      10     6
stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/libSystem.B.dylib\0", 0x16CFF9E10, 0x0)
= -1 Err#2

1727/0x31f7:      1709     104     97 open("/dev/dtracehelper\0", 0x2, 0x0) = 3 0

1728/0x31f8:      1645     187     56 open("/dev/dtracehelper\0", 0x2, 0x0) = 3 0

1728/0x31f8:      2159     829    513 ioctl(0x3, 0x80086804, 0x16CFF9A58) = 0 0

1728/0x31f8:      2165      7      4 close(0x3) = 0 0

1727/0x31f7:      2140    1205    419 ioctl(0x3, 0x80086804, 0x16B9B1A58) = 0 0

1727/0x31f7:      2145      5      3 close(0x3) = 0 0

1728/0x31f8:      2771     47     44 open("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client/lab_3_client\0", 0x0, 0x0) = 3 0

1728/0x31f8:      2778      7      5 __mac_syscall(0x18ADCDAF, 0x2, 0x16CFF9160) =
0 0

1728/0x31f8:      2809     28     26 map_with_linking_np(0x16CFF8CF0, 0x1, 0x16CFF8D20) =
0 0

```

| | | | | | |
|-----|---|------|----|--|---------------|
| | 1728/0x31f8: | 2811 | 3 | 1 close(0x3) | = 0 0 |
| | 1728/0x31f8: | 2817 | 5 | 3 mprotect(0x102E18000, 0x4000, 0x1) | = 0 0 |
| | 1728/0x31f8: = 0 0 | 2849 | 3 | 0 shared_region_check_np(0xFFFFFFFFFFFFFFFF, 0x0, 0x0) | |
| | 1727/0x31f7: build-release/src/lab-3/client/lab_3_client\0", 0x0, 0x0) | 2597 | 26 | 24 open("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake- = 3 0 | |
| | 1728/0x31f8: | 2852 | 3 | 1 mprotect(0x103230000, 0x4000, 0x1) | = 0 0 |
| 0 0 | 1727/0x31f7: | 2601 | 5 | 3 __mac_syscall(0x18ADCDAF, 0x2, 0x16B9B1160) | = |
| | 1728/0x31f8: = -1 Err#2 | 2876 | 6 | 3 access("/AppleInternal/XBS/.isChrooted\0", 0x0, 0x0) | |
| 0 0 | 1727/0x31f7: | 2615 | 14 | 11 map_with_linking_np(0x16B9B0CF0, 0x1, 0x16B9B0D20) | = |
| | 1727/0x31f7: | 2617 | 2 | 1 close(0x3) | = 0 0 |
| | 1727/0x31f7: | 2619 | 3 | 1 mprotect(0x104460000, 0x4000, 0x1) | = 0 0 |
| | 1728/0x31f8: = 1073746399 0 | 2907 | 5 | 2 bsdthread_register(0x18B0D0D2C, 0x18B0D0D20, 0x4000) | |
| | 1727/0x31f7: = 0 0 | 2646 | 3 | 0 shared_region_check_np(0xFFFFFFFFFFFFFFFF, 0x0, 0x0) | |
| | 1727/0x31f7: | 2650 | 3 | 1 mprotect(0x104878000, 0x4000, 0x1) | = 0 0 |
| | 1727/0x31f7: = -1 Err#2 | 2667 | 4 | 2 access("/AppleInternal/XBS/.isChrooted\0", 0x0, 0x0) | |
| | 1728/0x31f8: | 2961 | 2 | 0 getpid(0x0, 0x0, 0x0) | = 1728 0 |
| 3 0 | 1728/0x31f8: | 2968 | 7 | 5 shm_open(0x18AF6BF51, 0x0, 0xFFFFFFFF8B08F388) | = |
| | 1727/0x31f7: = 1073746399 0 | 2689 | 3 | 0 bsdthread_register(0x18B0D0D2C, 0x18B0D0D20, 0x4000) | |
| | 1728/0x31f8: | 2971 | 3 | 1 fstat64(0x3, 0x16CFF9DF0, 0x0) | = 0 0 |
| 0 | 1728/0x31f8: | 2978 | 11 | 6 mmap(0x0, 0x4000, 0x1, 0x40001, 0x3, 0x0) | = 0x10327C000 |
| | 1728/0x31f8: | 2983 | 2 | 1 close(0x3) | = 0 0 |
| | 1727/0x31f7: | 2723 | 2 | 0 getpid(0x0, 0x0, 0x0) | = 1727 0 |
| 3 0 | 1727/0x31f7: | 2727 | 5 | 3 shm_open(0x18AF6BF51, 0x0, 0xFFFFFFFF8B08F388) | = |
| | 1728/0x31f8: | 3001 | 4 | 2 ioctl(0x2, 0x4004667A, 0x16CFF9E9C) | = 0 0 |
| | 1727/0x31f7: | 2728 | 2 | 0 fstat64(0x3, 0x16B9B1DF0, 0x0) | = 0 0 |
| 0 | 1727/0x31f7: | 2732 | 6 | 3 mmap(0x0, 0x4000, 0x1, 0x40001, 0x3, 0x0) | = 0x1048C4000 |
| | 1728/0x31f8: | 3014 | 3 | 1 mprotect(0x103288000, 0x4000, 0x0) | = 0 0 |
| | 1727/0x31f7: | 2735 | 2 | 0 close(0x3) | = 0 0 |
| | 1728/0x31f8: | 3016 | 2 | 1 mprotect(0x103294000, 0x4000, 0x0) | = 0 0 |
| | 1728/0x31f8: | 3024 | 2 | 1 mprotect(0x103298000, 0x4000, 0x0) | = 0 0 |
| | 1727/0x31f7: | 2747 | 2 | 1 ioctl(0x2, 0x4004667A, 0x16B9B1E9C) | = 0 0 |
| | 1728/0x31f8: | 3025 | 2 | 0 mprotect(0x1032A4000, 0x4000, 0x0) | = 0 0 |
| | 1727/0x31f7: | 2760 | 3 | 1 mprotect(0x1048D0000, 0x4000, 0x0) | = 0 0 |
| | 1728/0x31f8: | 3032 | 2 | 0 mprotect(0x1032A8000, 0x4000, 0x0) | = 0 0 |
| | 1727/0x31f7: | 2761 | 2 | 0 mprotect(0x1048DC000, 0x4000, 0x0) | = 0 0 |

| | | | | |
|--------------|------|-----|---|-------|
| 1728/0x31f8: | 3034 | 2 | 0 mprotect(0x1032B4000, 0x4000, 0x0) | = 0 0 |
| 1727/0x31f7: | 2769 | 2 | 1 mprotect(0x1048E0000, 0x4000, 0x0) | = 0 0 |
| 1728/0x31f8: | 3042 | 2 | 1 mprotect(0x103280000, 0xA0, 0x1) | = 0 0 |
| 1727/0x31f7: | 2770 | 2 | 0 mprotect(0x1048EC000, 0x4000, 0x0) | = 0 0 |
| 1728/0x31f8: | 3044 | 3 | 1 mprotect(0x103280000, 0xA0, 0x3) | = 0 0 |
| 1727/0x31f7: | 2777 | 2 | 0 mprotect(0x1048F0000, 0x4000, 0x0) | = 0 0 |
| 1728/0x31f8: | 3050 | 2 | 0 mprotect(0x103280000, 0xA0, 0x1) | = 0 0 |
| 1728/0x31f8: | 3055 | 2 | 0 mprotect(0x1032B8000, 0x4000, 0x1) | = 0 0 |
| 1728/0x31f8: | 3062 | 3 | 1 mprotect(0x1032BC000, 0xA0, 0x1) | = 0 0 |
| 1727/0x31f7: | 2792 | 40 | 15 mprotect(0x1048FC000, 0x4000, 0x0) | = 0 0 |
| 1727/0x31f7: | 2804 | 4 | 1 mprotect(0x1048C8000, 0xA0, 0x1) | = 0 0 |
| 1727/0x31f7: | 2807 | 3 | 1 mprotect(0x1048C8000, 0xA0, 0x3) | = 0 0 |
| 1727/0x31f7: | 2811 | 2 | 1 mprotect(0x1048C8000, 0xA0, 0x1) | = 0 0 |
| 1727/0x31f7: | 2818 | 2 | 1 mprotect(0x104900000, 0x4000, 0x1) | = 0 0 |
| 1727/0x31f7: | 2825 | 3 | 1 mprotect(0x104904000, 0xA0, 0x1) | = 0 0 |
| 1728/0x31f8: | 3078 | 4 | 2 mprotect(0x1032BC000, 0xA0, 0x3) | = 0 0 |
| 1727/0x31f7: | 2835 | 3 | 1 mprotect(0x104904000, 0xA0, 0x3) | = 0 0 |
| 1728/0x31f8: | 3083 | 3 | 1 mprotect(0x1032BC000, 0xA0, 0x1) | = 0 0 |
| 1727/0x31f7: | 2841 | 3 | 1 mprotect(0x104904000, 0xA0, 0x1) | = 0 0 |
| 1728/0x31f8: | 3086 | 3 | 2 mprotect(0x103280000, 0xA0, 0x3) | = 0 0 |
| 1727/0x31f7: | 2843 | 3 | 1 mprotect(0x1048C8000, 0xA0, 0x3) | = 0 0 |
| 1728/0x31f8: | 3090 | 3 | 1 mprotect(0x103280000, 0xA0, 0x1) | = 0 0 |
| 1727/0x31f7: | 2846 | 2 | 1 mprotect(0x1048C8000, 0xA0, 0x1) | = 0 0 |
| 1727/0x31f7: | 2855 | 28 | 3 mprotect(0x104900000, 0x4000, 0x3) | = 0 0 |
| 1727/0x31f7: | 2858 | 2 | 0 mprotect(0x104900000, 0x4000, 0x1) | = 0 0 |
| 1727/0x31f7: | 2861 | 1 | 0 mprotect(0x104878000, 0x40000, 0x3) | = 0 0 |
| 1727/0x31f7: | 2867 | 1 | 0 mprotect(0x104878000, 0x40000, 0x1) | = 0 0 |
| 1727/0x31f7: | 2879 | 1 | 0 objc_bp_assist_cfg_np(0x18ACFD000, 0x80000018001C1048, 0x0) | |
| = -1 Err#5 | | | | |
| 1728/0x31f8: | 3101 | 81 | 6 mprotect(0x1032B8000, 0x4000, 0x3) | = 0 0 |
| 1728/0x31f8: | 3106 | 2 | 0 mprotect(0x1032B8000, 0x4000, 0x1) | = 0 0 |
| 1728/0x31f8: | 3108 | 2 | 1 mprotect(0x103230000, 0x40000, 0x3) | = 0 0 |
| 1728/0x31f8: | 3116 | 2 | 1 mprotect(0x103230000, 0x40000, 0x1) | = 0 0 |
| 1728/0x31f8: | 3130 | 3 | 0 objc_bp_assist_cfg_np(0x18ACFD000, 0x80000018001C1048, 0x0) | |
| = -1 Err#5 | | | | |
| 1728/0x31f8: | 3134 | 2 | 0 issetugid(0x0, 0x0, 0x0) | = 0 0 |
| 1728/0x31f8: | 3172 | 2 | 1 mprotect(0x103230000, 0x40000, 0x3) | = 0 0 |
| 1728/0x31f8: | 3239 | 8 | 6 getentropy(0x16CFF9508, 0x20, 0x0) | = 0 0 |
| 1727/0x31f7: | 2887 | 303 | 3 issetugid(0x0, 0x0, 0x0) | = 0 0 |
| 1727/0x31f7: | 2918 | 3 | 1 mprotect(0x104878000, 0x40000, 0x3) | = 0 0 |
| 1728/0x31f8: | 3292 | 27 | 1 mprotect(0x103230000, 0x40000, 0x1) | = 0 0 |
| 1727/0x31f7: | 2965 | 3 | 1 getentropy(0x16B9B1508, 0x20, 0x0) | = 0 0 |

```

1728/0x31f8:      3319      1      0 mprotect(0x103230000, 0x40000, 0x3)      = 0 0

1728/0x31f8:      3323      1      0 mprotect(0x103230000, 0x40000, 0x1)      = 0 0

1728/0x31f8:      3344      18     17 getattrlist("/Users/skidunion/Work/Depot/Personal/mai-os-3-
sem/cmake-build-release/src/lab-3/server/../client/lab_3_client\0", 0x16CFF9D80, 0x16CFF9D98)
0 0      =

1728/0x31f8:      3370      25     24 access("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client\0", 0x4, 0x0)      = 0 0

1727/0x31f7:      3033      3      1 mprotect(0x104878000, 0x40000, 0x1)      = 0 0

1728/0x31f8:      3382      12     11 open("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client\0", 0x0, 0x0)      = 3 0

1728/0x31f8:      3384      2      1 fstat64(0x3, 0x159E04450, 0x0)      = 0 0

1728/0x31f8:      3385      1      0 csrctl(0x0, 0x16CFF9FAC, 0x4)      = 0 0

1728/0x31f8:      3388      3      2 fcntl(0x3, 0x32, 0x16CFF9C68)      = 0 0

1728/0x31f8:      3390      2      1 close(0x3)      = 0 0

1727/0x31f7:      3070      3      1 mprotect(0x104878000, 0x40000, 0x3)      = 0 0

1727/0x31f7:      3076      2      1 mprotect(0x104878000, 0x40000, 0x1)      = 0 0

1728/0x31f8:      3406      13     12 open("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client/Info.plist\0", 0x0, 0x0)      = -1 Err#2

1728/0x31f8:      3421      3      2 proc_info(0x2, 0x6C0, 0xD)      = 64 0

1727/0x31f7:      3104      24     22 getattrlist("/Users/skidunion/Work/Depot/Personal/mai-os-3-
sem/cmake-build-release/src/lab-3/server/../client/lab_3_client\0", 0x16B9B1D80, 0x16B9B1D98)
0 0      =

1728/0x31f8:      3438      3      2 csops_audittoken(0x6C0, 0x10, 0x16CFF9FF0)      = 0 0

1728/0x31f8:      3453      10     7 sysctl([unknown, 3, 0, 0, 0, 0] (2), 0x16CFFA348, 0x16CFFA340,
0x18E502D3D, 0x15) = 0 0

1727/0x31f7:      3137      32     30 access("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client\0", 0x4, 0x0)      = 0 0

1728/0x31f8:      3455      2      1 sysctl([CTL_KERN, 138, 0, 0, 0, 0] (2), 0x16CFFA3D8, 0x16CFFA3D0,
0x0, 0x0)      = 0 0

1728/0x31f8:      3458      1      0 csops(0x6C0, 0x0, 0x16CFFA47C)      = 0 0

1728/0x31f8:      3462      1      0 mprotect(0x103230000, 0x40000, 0x3)      = 0 0

1727/0x31f7:      3163      26     24 open("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client\0", 0x0, 0x0)      = 3 0

1727/0x31f7:      3166      5      2 fstat64(0x3, 0x12DE04450, 0x0)      = 0 0

1727/0x31f7:      3168      2      0 csrctl(0x0, 0x16B9B1FAC, 0x4)      = 0 0

1727/0x31f7:      3171      4      2 fcntl(0x3, 0x32, 0x16B9B1C68)      = 0 0

1727/0x31f7:      3175      4      2 close(0x3)      = 0 0

1727/0x31f7:      3188      6      5 open("/Users/skidunion/Work/Depot/Personal/mai-os-3-sem/cmake-
build-release/src/lab-3/client/Info.plist\0", 0x0, 0x0)      = -1 Err#2

1727/0x31f7:      3209      5      2 proc_info(0x2, 0x6BF, 0xD)      = 64 0

1727/0x31f7:      3234      5      2 csops_audittoken(0x6BF, 0x10, 0x16B9B1FF0)      = 0 0

1728/0x31f8:      3583      6      5 sem_open(0x16CFFBAF9, 0x0, 0x0)      = 3 0

1728/0x31f8:      3585      2      1 sem_open(0x16CFFBB05, 0x0, 0x0)      = 10 0

1727/0x31f7:      3254      11     7 sysctl([unknown, 3, 0, 0, 0, 0] (2), 0x16B9B2348, 0x16B9B2340,
0x18E502D3D, 0x15) = 0 0

1728/0x31f8:      3588      3      2 shm_open(0x16CFFBAE7, 0x2, 0x180)      = 11 0

1728/0x31f8:      3592      4      3 mmap(0x0, 0x400, 0x3, 0x40001, 0xB, 0x0)      = 0x102E2C000

```

```

1727/0x31f7:      3257      4      2 sysctl([CTL_KERN, 138, 0, 0, 0, 0] (2), 0x16B9B23D8, 0x16B9B23D0,
0x0, 0x0)          = 0 0

1727/0x31f7:      3261      3      1 csops(0x6BF, 0x0, 0x16B9B247C)          = 0 0
1727/0x31f7:      3266      3      1 mprotect(0x104878000, 0x40000, 0x3)          = 0 0
1727/0x31f7:      3361      9      7 sem_open(0x16B9B3AF9, 0x0, 0x0)          = 3 0
1727/0x31f7:      3364      4      2 sem_open(0x16B9B3B05, 0x0, 0x0)          = 10 0
1727/0x31f7:      3369      5      3 shm_open(0x16B9B3AE7, 0x2, 0x180)          = 11 0
1727/0x31f7:      3374      7      4 mmap(0x0, 0x400, 0x3, 0x40001, 0xB, 0x0)          = 0x104474000
0

```

```

1 ABABABABABABBA
2 BABABABABABABA
3 JKASJDKSDJIWQEJ
4 kjsadkfjasflkja
5 ababababbaba
6 babababababba
7 ksadjfdksajfiger
8 c
9 a
10 b

```

```

1660/0x2f04:      4169 29213523      42 read(0x0, "1 ABABABABABABBA\n\0", 0x1000)          = 17 0
1660/0x2f04:      4398      81      6 sem_post(0x6, 0x0, 0x0)          = 0 0
1727/0x31f7:      3382 29206204      7 sem_wait(0x3, 0x0, 0x0)          = 0 0
1727/0x31f7:      3465      6      3 sem_post(0xA, 0x0, 0x0)          = 0 0
1660/0x2f04:      4404      211      5 sem_wait(0x7, 0x0, 0x0)          = 0 0
1660/0x2f04:      4454      122     47 write(0x4, "1BBBBBBB\n\0", 0xA)          = 10 0
1660/0x2f04:      4478      7      4 read(0x0, "2 BABABABABABABA\n\0", 0x1000)          = 17 0
1660/0x2f04:      4484      7      4 sem_post(0x8, 0x0, 0x0)          = 0 0
1728/0x31f8:      3596 29206805      4 sem_wait(0x3, 0x0, 0x0)          = 0 0
1728/0x31f8:      3617      5      3 sem_post(0xA, 0x0, 0x0)          = 0 0
1660/0x2f04:      4489      31      4 sem_wait(0x9, 0x0, 0x0)          = 0 0
1660/0x2f04:      4513      25     23 write(0x5, "2BBBBBBB\n\0", 0xA)          = 10 0
1660/0x2f04:      4518      4      3 read(0x0, "3 JKASJDKSDJIWQEJ\n\0", 0x1000)          = 18 0
1660/0x2f04:      4522      5      3 sem_post(0x6, 0x0, 0x0)          = 0 0
1727/0x31f7:      3470     284      4 sem_wait(0x3, 0x0, 0x0)          = 0 0
1727/0x31f7:      3473      4      2 sem_post(0xA, 0x0, 0x0)          = 0 0
1660/0x2f04:      4526      12      3 sem_wait(0x7, 0x0, 0x0)          = 0 0
1660/0x2f04:      4537      11     10 write(0x4, "3 JKSDJWQJ\n\0", 0xF)          = 15 0
1660/0x2f04:      4540      4      2 read(0x0, "4 kjsadkfjasflkja\n\0", 0x1000)          = 18 0
1660/0x2f04:      4544      4      2 sem_post(0x8, 0x0, 0x0)          = 0 0
1728/0x31f8:      3620     122      3 sem_wait(0x3, 0x0, 0x0)          = 0 0
1728/0x31f8:      3625      4      3 sem_post(0xA, 0x0, 0x0)          = 0 0
1660/0x2f04:      4548      12      3 sem_wait(0x9, 0x0, 0x0)          = 0 0

```

| | | | | | |
|--------------|------|---------|----|---|--------|
| 1660/0x2f04: | 4555 | 8 | 6 | write(0x5, "4 kjsdkfjsflkj\n\0", 0xF) | = 15 0 |
| 1660/0x2f04: | 4558 | 3 | 2 | read(0x0, "5 ababababbaba\n\0", 0x1000) | = 15 0 |
| 1660/0x2f04: | 4562 | 4 | 2 | sem_post(0x6, 0x0, 0x0) | = 0 0 |
| 1727/0x31f7: | 3478 | 105 | 4 | sem_wait(0x3, 0x0, 0x0) | = 0 0 |
| 1727/0x31f7: | 3483 | 5 | 3 | sem_post(0xA, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4566 | 15 | 3 | sem_wait(0x7, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4572 | 7 | 5 | write(0x4, "5 bbbbbb\n\0", 0x9) | = 9 0 |
| 1660/0x2f04: | 4575 | 3 | 2 | read(0x0, "6 babababababba\n\0", 0x1000) | = 16 0 |
| 1660/0x2f04: | 4579 | 5 | 3 | sem_post(0x8, 0x0, 0x0) | = 0 0 |
| 1728/0x31f8: | 3631 | 104 | 5 | sem_wait(0x3, 0x0, 0x0) | = 0 0 |
| 1728/0x31f8: | 3639 | 75 | 5 | sem_post(0xA, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4584 | 92 | 3 | sem_wait(0x9, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4591 | 9 | 7 | write(0x5, "6 bbbbbb\n\0", 0xA) | = 10 0 |
| 1660/0x2f04: | 4595 | 4 | 2 | read(0x0, "7 ksadjfksajfiqr\n\0", 0x1000) | = 19 0 |
| 1660/0x2f04: | 4599 | 4 | 2 | sem_post(0x6, 0x0, 0x0) | = 0 0 |
| 1727/0x31f7: | 3489 | 180 | 6 | sem_wait(0x3, 0x0, 0x0) | = 0 0 |
| 1727/0x31f7: | 3494 | 5 | 2 | sem_post(0xA, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4602 | 15 | 2 | sem_wait(0x7, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4609 | 8 | 6 | write(0x4, "7 ksadjfksjqr\n\0", 0xF) | = 15 0 |
| 1660/0x2f04: | 4611 | 2 | 1 | read(0x0, "8 c\n\0", 0x1000) | = 4 0 |
| 1660/0x2f04: | 4614 | 4 | 2 | sem_post(0x8, 0x0, 0x0) | = 0 0 |
| 1728/0x31f8: | 3644 | 100 | 4 | sem_wait(0x3, 0x0, 0x0) | = 0 0 |
| 1728/0x31f8: | 3648 | 4 | 2 | sem_post(0xA, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4618 | 12 | 3 | sem_wait(0x9, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4628 | 10 | 8 | write(0x5, "8 c\n\0", 0x4) | = 4 0 |
| 1660/0x2f04: | 4630 | 3 | 1 | read(0x0, "9 a\n\0", 0x1000) | = 4 0 |
| 1660/0x2f04: | 4634 | 4 | 2 | sem_post(0x6, 0x0, 0x0) | = 0 0 |
| 1727/0x31f7: | 3498 | 95 | 3 | sem_wait(0x3, 0x0, 0x0) | = 0 0 |
| 1727/0x31f7: | 3502 | 4 | 2 | sem_post(0xA, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4637 | 11 | 2 | sem_wait(0x7, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4645 | 8 | 7 | write(0x4, "9 \n\0", 0x3) | = 3 0 |
| 1660/0x2f04: | 4647 | 2 | 1 | read(0x0, "10 b\n\0", 0x1000) | = 5 0 |
| 1660/0x2f04: | 4651 | 5 | 3 | sem_post(0x8, 0x0, 0x0) | = 0 0 |
| 1728/0x31f8: | 3653 | 96 | 4 | sem_wait(0x3, 0x0, 0x0) | = 0 0 |
| 1728/0x31f8: | 3656 | 4 | 2 | sem_post(0xA, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4655 | 12 | 2 | sem_wait(0x9, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4660 | 6 | 4 | write(0x5, "10 b\n\0", 0x5) | = 5 0 |
| 1660/0x2f04: | 4673 | 757991 | 11 | read(0x0, "\n\0", 0x1000) | = 1 0 |
| 1660/0x2f04: | 4695 | 6998724 | 15 | read(0x0, "\367\001\0", 0x1000) | = 0 0 |
| 1660/0x2f04: | 4723 | 112 | 14 | sem_post(0x6, 0x0, 0x0) | = 0 0 |

| | | | | | |
|--------------|------|---------|----|--------------------------------------|-------------|
| 1727/0x31f7: | 3509 | 7756966 | 6 | sem_wait(0x3, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4729 | 7 | 4 | sem_post(0x8, 0x0, 0x0) | = 0 0 |
| 1728/0x31f8: | 3663 | 7756945 | 6 | sem_wait(0x3, 0x0, 0x0) | = 0 0 |
| 1728/0x31f8: | 3688 | 11 | 8 | munmap(0x102E2C000, 0x400) | = 0 0 |
| 1728/0x31f8: | 3695 | 8 | 5 | sem_close(0x3, 0x0, 0x0) | = 0 0 |
| 1728/0x31f8: | 3697 | 3 | 1 | sem_close(0xA, 0x0, 0x0) | = 0 0 |
| 1727/0x31f7: | 3535 | 97 | 8 | munmap(0x104474000, 0x400) | = 0 0 |
| 1727/0x31f7: | 3538 | 4 | 2 | sem_close(0x3, 0x0, 0x0) | = 0 0 |
| 1727/0x31f7: | 3539 | 2 | 0 | sem_close(0xA, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4746 | 644 | 13 | wait4(0xFFFFFFFFFFFFFFFF, 0x0, 0x0) | = 1728 0 |
| 1660/0x2f04: | 4757 | 48 | 6 | wait4(0xFFFFFFFFFFFFFFFF, 0x0, 0x0) | = 1727 0 |
| 1660/0x2f04: | 4757 | 1 | 0 | wait4(0xFFFFFFFFFFFFFFFF, 0x0, 0x0) | = -1 Err#10 |
| 1660/0x2f04: | 4838 | 47 | 41 | close(0x5) | = 0 0 |
| 1660/0x2f04: | 4840 | 3 | 1 | sem_close(0x8, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4850 | 11 | 8 | sem_unlink(0x600001A08050, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4851 | 2 | 0 | sem_close(0x9, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4853 | 3 | 1 | sem_unlink(0x600001A08070, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4864 | 11 | 10 | close(0x4) | = 0 0 |
| 1660/0x2f04: | 4866 | 2 | 0 | sem_close(0x6, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4869 | 4 | 2 | sem_unlink(0x600001A08008, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4870 | 2 | 0 | sem_close(0x7, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4871 | 3 | 1 | sem_unlink(0x600001A08028, 0x0, 0x0) | = 0 0 |
| 1660/0x2f04: | 4916 | 9 | 7 | munmap(0x1043B0000, 0x400) | = 0 0 |
| 1660/0x2f04: | 4922 | 7 | 5 | shm_unlink(0x16BA776A0, 0x0, 0x0) | = 0 0 |

| CALL | COUNT |
|-----------------------|-------|
| ftruncate | 1 |
| shm_unlink | 1 |
| execve | 2 |
| fork | 2 |
| csops | 3 |
| csops_audittoken | 3 |
| exit | 3 |
| fstatat64 | 3 |
| getentropy | 3 |
| issetugid | 3 |
| map_with_linking_np | 3 |
| objc_bp_assist_cfg_np | 3 |
| wait4 | 3 |
| sem_unlink | 4 |
| bsdthread_register | 5 |

| | |
|------------------------|----|
| crossarch_trap | 5 |
| getpid | 5 |
| proc_info | 5 |
| access | 6 |
| fstat64 | 6 |
| getattrlist | 6 |
| getfsstat64 | 6 |
| ioctl | 6 |
| openat | 6 |
| shm_open | 6 |
| thread_selfid | 6 |
| sem_close | 8 |
| sem_open | 8 |
| shared_region_check_np | 8 |
| csrctl | 9 |
| dup | 9 |
| fsgetpath | 9 |
| mmap | 9 |
| __mac_syscall | 12 |
| sysctl | 12 |
| write | 12 |
| read | 14 |
| fcntl | 15 |
| munmap | 21 |
| sem_post | 22 |
| sem_wait | 22 |
| open | 23 |
| stat64 | 33 |
| close | 38 |
| mprotect | 84 |

Вывод

В ходе выполнения этой лабораторной работы я познакомился с концепцией Shared Memory и Memory Mapping в POSIX. Я научился применять семафоры для синхронизации нескольких процессов. Написал этот код я не с первого раза: первой ошибкой был use-after-free из-за стандартного конструктора копирования у обёртки SharedMemory, которая уничтожала свой ресурс после выхода из области видимости. Второй ошибкой были неправильные аргументы к sem_open — изначально семафор был разблокирован, из-за чего была гонка между процессами. В целом, работа мне показалась несложной, однако из-за таких нюансов её отладка заняла много времени.