Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу
«Операционные системы»

Студент: Маринин И.С.
Группа: М8О–208Б–20
Преподаватель: Миронов Е.С.
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2021.

Постановка задачи

Цель работы

Приобретение практических навыков диагностики работы программного обеспечения на примере 3 лабораторной работы.

Задание

Провести диагностику работы 3 лабораторной работы при помощи dtruss (аналог strace на MacOS), объяснить результат работы dtruss.

Вывод dtruss

```
        PID/THRD  RELATIVE  ELAPSD    CPU SYSCALL(args)                 = return
Sitax: ./a.out Threads_num Num_for_test
15200/0x5f3a4:      782:       0:      0 fork()                  = 0 2
15200/0x5f3a4:      878     187    95 open("/dev/dtracehelper\0", 0x2, 0x0)        = 3 0
15200/0x5f3a4:     1103     223   221 ioctl(0x3, 0x80086804, 0x7FFEE34F39C0)          = 0 0
15200/0x5f3a4:     1119      17    14 close(0x3)             = 0 0
15200/0x5f3a4:     1198       6     2 mprotect(0x10C70F000, 0x4000, 0x1)         = 0 0
15200/0x5f3a4:     1246       5     2 access("/AppleInternal/XBS/.isChrooted\0", 0x0, 0x0)
= -1 2
15200/0x5f3a4:     1267       6     2 bsdthread_register(0x7FFF2030F434, 0x7FFF2030F420, 0x2000)
= 1073742047 0
15200/0x5f3a4:     1690       7     4 shm_open(0x7FFF201E7F66, 0x0, 0x201E6CBB)              = 3 0
15200/0x5f3a4:     1692       3     0 fstat64(0x3, 0x7FFEE34F2950, 0x0)          = 0 0
15200/0x5f3a4:     1698       9     4 mmap(0x0, 0x1000, 0x1, 0x40001, 0x3, 0x0)                  =
0x10C71D000 0
15200/0x5f3a4:     1701       2     0 close(0x3)              = 0 0
15200/0x5f3a4:     1925       3     1 ioctl(0x2, 0x4004667A, 0x7FFEE34F2A04) = 0 0
15200/0x5f3a4:     1936       3     1 mprotect(0x10C723000, 0x1000, 0x0)         = 0 0
15200/0x5f3a4:     1937       2     0 mprotect(0x10C72A000, 0x1000, 0x0)         = 0 0
15200/0x5f3a4:     1946       2     0 mprotect(0x10C72B000, 0x1000, 0x0)         = 0 0
15200/0x5f3a4:     1947       2     1 mprotect(0x10C732000, 0x1000, 0x0)         = 0 0
15200/0x5f3a4:     1959       3     1 mprotect(0x10C71E000, 0x90, 0x1) = 0 0
15200/0x5f3a4:     1965       3     0 mprotect(0x10C733000, 0x1000, 0x1)         = 0 0
15200/0x5f3a4:     1968       3     1 mprotect(0x10C71E000, 0x90, 0x3) = 0 0
15200/0x5f3a4:     1978       3     1 mprotect(0x10C71E000, 0x90, 0x1) = 0 0
15200/0x5f3a4:     2005       3     0 issetugid(0x0, 0x0, 0x0)                = 0 0
15200/0x5f3a4:     2309       5     1 getentropy(0x7FFEE34F21A0, 0x20, 0x0)    = 0 0
15200/0x5f3a4:     2312       4     2 getentropy(0x7FFEE34F21F0, 0x40, 0x0)    = 0 0
15200/0x5f3a4:     2389       3     0 getpid(0x0, 0x0, 0x0)                = 15200 0
15200/0x5f3a4:     2397       4     2 stat64("/AppleInternal\0", 0x7FFEE34F2F30, 0x0)              = -1 2
15200/0x5f3a4:     2414       5     1 csops_audittoken(0x3B60, 0x7, 0x7FFEE34F2A60)            = -1 22
```

15200/0x5f3a4: 2417 4 2 proc_info(0x2, 0x3B60, 0xD) = 64 0
15200/0x5f3a4: 2429 3 0 csops_audittoken(0x3B60, 0x7, 0x7FFEE34F2B50) = -1 22
15200/0x5f3a4: 2440 11 6 sysctlbyname(kern.osvariant_status, 0x15, 0x7FFEE34F2F80, 0x7FFEE34F2F78, 0x0) = 0 0
15200/0x5f3a4: 2442 2 0 csops(0x3B60, 0x0, 0x7FFEE34F2FB4) = 0 0
15200/0x5f3a4: 2447 5 2 sysctlbyname(kern.system_version_compat, 0x1A, 0x0, 0x0, 0x7FFEE34F2FEC) = 0 0
15200/0x5f3a4: 2517 64 0 getrlimit(0x1008, 0x7FFEE34F47C0, 0x0) = 0 0
15200/0x5f3a4: 2521 5 3 fstat64(0x1, 0x7FFEE34F47A8, 0x0) = 0 0
15200/0x5f3a4: 2524 3 0 ioctl(0x1, 0x4004667A, 0x7FFEE34F47F4) = 0 0
dtrace: error on enabled probe ID 1682 (ID 956: syscall::write_nocancel:return): invalid kernel access in action #13 at DIF offset 6

CALL	COUNT
access	1
bsdthread_register	1
csops	1
exit	1
getpid	1
getrlimit	1
issetugid	1
mmap	1
open	1
proc_info	1
shm_open	1
stat64	1
write_nocancel	1
close	2
csops_audittoken	2
fstat64	2
getentropy	2
sysctlbyname	2
ioctl	3
mprotect	9

Описание работы

open("/dev/dtracehelper\0", 0x2, 0x0) = 3 0

Преобразует путь к файлу в описатель файла, первый аргумент - путь, второй - флаг, третий - мод. Open возвращает новый описатель файла или -1 в случае ошибки.

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

Проверяет /AppleInternal/XBS/.isChrooted\0 на существование и на наличие прав на чтение или запись, возвращает -1 – или не существует /AppleInternal/XBS/.isChrooted\0 или нет прав на чтение или запись.

fstat64(0x3, 0x7FFEE34F2950, 0x0) = 0 0

Заполняет структуру указанную вторым аргументом fstat информацией об файле с файловым дескриптором 0x3. Возвращает 0 – успешное выполнение.

mmap(0x0, 0x1000, 0x1, 0x40001, 0x3, 0x0) = 0x10C71D000 0

Создает отображение файла с файловым дескриптором 0x3 в память, начиная с адреса 0x0 (система сама выбирает), размер = 0x1000 байт, с правами защиты памяти на чтение 0x1, задает тип отражаемого объекта 0x40001 - создает неразделимое отражение с механизмом copy-on-write, запись в эту область памяти не влияет на файл, не определено, являются или нет изменения в файле после вызова mmap видимыми в отраженном диапазоне. Возвращает указатель на начало отраженной памяти 0x10C71D000.

close(0x3) = 0 0
Закрывает файл с файловым дескриптором 0x3. Возвращает 0 – успешное выполнение.

mprotect(0x10C70F000, 0x4000, 0x1) = 0 0

Контролирует доступ к области памяти начинающейся с адресса 0x10C70F000 длины 0x4000 байт, доступ к памяти запрещен - 0x1. Если программой производится запрещенный этой функцией доступ к памяти, то такая программа получает сигнал SIGSEGV. Возвращает 0 – успешное завершение.

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

Возвращает 1, если среда процесса или адресное пространство памяти считаются испорченными, и возвращает 0 в противном случае

ioctl(0x3, 0x80086804, 0x7FFEE34F39C0) = 0 0

Это средство управления аппаратными устройствами. Первым аргументом функции(0x3) является дескриптор файла того устройства, которым требуется управлять. Второй аргумент (0x80086804)— это код запроса, обозначающего выполняемую операцию.

Вывод

Dtruss позволяет просматривать различные системные вызовы, которые происходят при работе программы. Благодаря этому можно удобно анализировать работу программы и искать ошибки.