

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: Матвеев Данил
Группа: М8О-207Б-21
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2023

Содержание

1. Репозиторий
2. Постановка задачи
3. Описание работы strace
4. Демонстрация работы strace
5. Вывод

Репозиторий

<https://github.com/MrDenli/OsLabs>

Постановка задачи

Подробно рассказать о каждом системном вызове из утилиты strace на примере лабораторной работы №3.

Описание работы strace

execve — открывает исполняемый файл

brk — определение конца сегмента данных для процесса

arch_prctl - задаёт состояние процесса или нити, зависящие от архитектуры

openat — открывает файл

fstat - считывает состояние файла

mmap, munmap - отражает файлы или устройства в памяти, снимает их отражение

mprotect - контролирует доступ к области памяти

read, write — чтение и запись

ПОДРОБНЕЕ:

`void *mmap(void *addr, size_t length, int prot, int flags, int fd, off_t offset);` - возвращает указатель на начало выделенного блока памяти. `Addr` — позволяет выбрать конкретный адрес, `length` — длина участка, `int prot` — разрешения (`write`, `read`), `fd` — файловый дескриптор, `offset` — сдвиг относительно адреса.

`getpid()` - возвращает `id` процесса, в котором была вызвана

`off_t lseek(int fd, off_t offset, int whence);` - сдвигает позицию в `fd` на значение `offset` в направлении `whence`.

`int execve(const char *pathname, char *const argv[], char *const envp[])` - выполняет программу

1) `pathname` — путь, ссылающийся на программу.

2) `argv` — это массив строк аргументов, передаваемых новой программе.

3) `envp` — это массив строк, обычно в форме `ключ=значение`, которые передаются в качестве среды новой программе.

`int access(const char *pathname, int mode)` - проверяет права пользователя на файл

1) `pathname` — путь, ссылающийся на программу. Если имя пути является символической ссылкой, оно разыменовывается.

2) `mode` - указывает проверки доступности, которые должны быть выполнены, и является либо значением `F_OK`, либо маской, состоящей из побитового ИЛИ одного или нескольких из `R_OK`, `W_OK` и `X_OK`. `F_OK` тесты для существования файла. `R_OK`, `W_OK` и `X_OK` проверяют, существует ли файл, и предоставляют права на чтение, запись и выполнение соответственно.

`ssize_t read(int fd, void *buf, size_t count)` - чтение из файлового дескриптора

1) `fd` — файловый дескриптор, из которого `read()` пытается прочесть.

3) `buf` – буффер, в который `read()` читает байты.

2) lenght — длина байтов, до которого обрезается файл.

```

execve("/main", ["/main"], 0x7ffe105f6db0 /* 45 vars */) = 0
brk(NULL) = 0x5652d5a0f000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffcda815c0) = -1 EINVAL (Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f0c425c8000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=59555, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 59555, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f0c425b9000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0\0....", 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2252096, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 2267328, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0c42200000
mmap(0x7f0c4229a000, 1114112, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9a000) = 0x7f0c4229a000
mmap(0x7f0c423aa000, 454656, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1aa000) = 0x7f0c423aa000
mmap(0x7f0c42419000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x218000) = 0x7f0c42419000
mmap(0x7f0c42427000, 10432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f0c42427000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0\0....", 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=125488, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 127720, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0c42599000
mmap(0x7f0c4259c000, 94208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f0c4259c000
mmap(0x7f0c425b3000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000) = 0x7f0c425b3000
mmap(0x7f0c425b7000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d000) = 0x7f0c425b7000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0\0....", 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0....", 784, 64) = 784
pread64(3, "\4\0\0\0\0\0\0\0\5\0\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0....", 48, 848) = 48
pread64(3, "\4\0\0\0\0\24\0\0\0\3\0\0\0GNU\0i8\235HZ\227\223\333\350s\360\352\223\340."..., 68, 896) = 68
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2216304, ...}, AT_EMPTY_PATH) = 0
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0....", 784, 64) = 784
mmap(NULL, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0c41e00000
mmap(0x7f0c41e28000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f0c41e28000
mmap(0x7f0c41fb0000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f0c41fb0000
mmap(0x7f0c42015000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x214000) = 0x7f0c42015000
mmap(0x7f0c4201b000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f0c4201b000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0\0....", 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0

```

```

mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0c424b2000
mmap(0x7f0c424c0000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7f0c424c0000
mmap(0x7f0c4253c000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x7f0c4253c000
mmap(0x7f0c42597000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x7f0c42597000
close(3) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f0c424b0000
arch_prctl(ARCH_SET_FS, 0x7f0c424b13c0) = 0
set_tid_address(0x7f0c424b1690) = 8165
set_robust_list(0x7f0c424b16a0, 24) = 0
rseq(0x7f0c424b1d60, 0x20, 0, 0x53053053) = 0
mprotect(0x7f0c42015000, 16384, PROT_READ) = 0
mprotect(0x7f0c42597000, 4096, PROT_READ) = 0
mprotect(0x7f0c425b7000, 4096, PROT_READ) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f0c424ae000
mprotect(0x7f0c42419000, 45056, PROT_READ) = 0
mprotect(0x5652d41b1000, 4096, PROT_READ) = 0
mprotect(0x7f0c42602000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f0c425b9000, 59555) = 0
getrandom("\xb7\x47\x9f\x8d\x32\xf3\x66\x6e", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x5652d5a0f000
brk(0x5652d5a30000) = 0x5652d5a30000
futex(0x7f0c4242777c, FUTEX_WAKE_PRIVATE, 2147483647) = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0
write(1, "1) \320\235\320\260\320\277\320\270\321\210\320\270\321\202\320\265 work, \320\265\321\201\320\273"..., 881)
Напишите work, если хотите переумножить матрицы
) = 88
write(1, "2) \320\222\320\262\320\265\320\264\320\270\321\202\320\265 test, \320\265\321\201\320\273\320\270"..., 892)
Введите test, если хотите провести тест скорости
) = 89
newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0
read(0, 0x5652d5a212c0, 1024) = ? ERESTARTSYS (To be restarted if SA_RESTART is set)
--- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
read(0, 0x5652d5a212c0, 1024) = ? ERESTARTSYS (To be restarted if SA_RESTART is set)
--- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---
read(0, ^Cstrace: Process 8165 detached
<detached ...>

```

Выводы

В данной лабораторной работе я использовал утилиту strace, для приобретения практических навыков диагностики работы программного обеспечения.