МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 "Компьютерные науки и прикладная математика"

Кафедра 806 "Вычислительная математика и программирование"

Лабораторная работа №8

По курсу «Операционные системы»

Студент: Степанов Н.Е.

Группа: М8О-208Б-23

Преподаватель: Миронов Е. С.

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

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

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2024

**Что такое strace?**

**strace** — это утилита для диагностики и отладки в операционных системах на базе Unix, которая позволяет отслеживать системные вызовы и сигналы, которые выполняет процесс. Она предоставляет информацию о взаимодействии программы с ядром операционной системы, включая вызовы файловых операций, сетевых взаимодействий и управления памятью. strace полезен для разработчиков и системных администраторов, поскольку помогает выявлять проблемы с производительностью, отладкой программ и анализом поведения приложений в реальном времени.

**Листинг работы strace и комментарии**

*execve("./lab3", ["./lab3", "data.txt"], 0x7fffd8f6f298 /\* 28 vars \*/) = 0*

**Программа успешно запущена с аргументом "data.txt".**

*brk(NULL) = 0x7fffdd2b6000*

*access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)*

*openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3*

*fstat(3, {st\_mode=S\_IFREG|0644, st\_size=44243, ...}) = 0*

*mmap(NULL, 44243, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7ff24f950000*

*close(3) = 0*

*openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = 3*

*read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@E\n\0\0\0\0\0"..., 832) = 832*

*fstat(3, {st\_mode=S\_IFREG|0644, st\_size=2526680, ...}) = 0*

*mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ff24f910000*

*mmap(NULL, 2544064, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff24f6a0000*

*mprotect(0x7ff24f73c000, 1835008, PROT\_NONE) = 0*

*mmap(0x7ff24f73c000, 1253376, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9c000) = 0x7ff24f73c000*

*mmap(0x7ff24f86e000, 577536, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ce000) = 0x7ff24f86e000*

*mmap(0x7ff24f8fc000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25b000) = 0x7ff24f8fc000*

*mmap(0x7ff24f90a000, 12736, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7ff24f90a000*

*close(3) = 0*

*openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3*

*read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\3006\0\0\0\0\0\0"..., 832) = 832*

*fstat(3, {st\_mode=S\_IFREG|0644, st\_size=146000, ...}) = 0*

*mmap(NULL, 148776, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff24f670000*

*mmap(0x7ff24f673000, 114688, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7ff24f673000*

*mmap(0x7ff24f68f000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1f000) = 0x7ff24f68f000*

*mmap(0x7ff24f693000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x22000) = 0x7ff24f693000*

*close(3) = 0*

*openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3*

*read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300A\2\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\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32*

*pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\7\2C\n\357\_\243\335\2449\206V>\237\374\304"..., 68, 880) = 68*

*fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2029592, ...}) = 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*

*pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32*

*pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\7\2C\n\357\_\243\335\2449\206V>\237\374\304"..., 68, 880) = 68*

*mmap(NULL, 2037344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff24f470000*

*mmap(0x7ff24f492000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x22000) = 0x7ff24f492000*

*mmap(0x7ff24f60a000, 319488, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19a000) = 0x7ff24f60a000*

*mmap(0x7ff24f658000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7ff24f658000*

*mmap(0x7ff24f65e000, 13920, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7ff24f65e000*

*close(3) = 0*

*openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libm.so.6", O\_RDONLY|O\_CLOEXEC) = 3*

*read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300\323\0\0\0\0\0\0"..., 832) = 832*

*fstat(3, {st\_mode=S\_IFREG|0644, st\_size=1369384, ...}) = 0*

*mmap(NULL, 1368336, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff24f321000*

*mmap(0x7ff24f32e000, 684032, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xd000) = 0x7ff24f32e000*

*mmap(0x7ff24f3d5000, 626688, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xb4000) = 0x7ff24f3d5000*

*mmap(0x7ff24f46e000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x14c000) = 0x7ff24f46e000*

*close(3) = 0*

*mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ff24f310000*

*arch\_prctl(ARCH\_SET\_FS, 0x7ff24f311100) = 0*

**Установка адреса сегмента FS для работы с потоками.**

*mprotect(0x7ff24f658000, 16384, PROT\_READ) = 0*

*mprotect(0x7ff24f46e000, 4096, PROT\_READ) = 0*

*mprotect(0x7ff24f693000, 4096, PROT\_READ) = 0*

*mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ff24f300000*

*mprotect(0x7ff24f8fc000, 45056, PROT\_READ) = 0*

*mprotect(0x7ff24f960000, 4096, PROT\_READ) = 0*

*mprotect(0x7ff24f94d000, 4096, PROT\_READ) = 0*

*munmap(0x7ff24f950000, 44243) = 0*

*brk(NULL) = 0x7fffdd2b6000*

*brk(0x7fffdd2d7000) = 0x7fffdd2d7000*

*openat(AT\_FDCWD, "data.txt", O\_RDONLY) = 3*

**Файл "data.txt" успешно открыт для чтения.**

*fstat(3, {st\_mode=S\_IFREG|0644, st\_size=5, ...}) = 0*

*mmap(NULL, 5, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7ff24f95a000*

**Файл "data.txt" успешно отображен в память для чтения.**

*openat(AT\_FDCWD, "result.txt", O\_WRONLY|O\_CREAT|O\_TRUNC, 0666) = 4*

**Файл "result.txt" успешно открыт для записи результата.**

*write(4, "\320\241\321\203\320\274\320\274\320\260 \321\207\320\270\321\201\320\265\320\273: 6\n", 25) = 25*

**Результат ("Сумма чисел: 6") успешно записан в файл "result.txt".**

*close(4) = 0*

*munmap(0x7ff24f95a000, 5) = 0*

*close(3) = 0*

*exit\_group(0) = ?*

**Программа завершена с кодом 0 (успешное выполнение).**

*+++ exited with 0 +++*

**Комментарии к листингу**

Вывод можно разделить на три основных этапа: запуск, выполнение и завершение.

**1. Запуск программы:**

Сначала с помощью различных системных вызовов инициализируется программа, подгружаются необходимые библиотеки (например, libstdc++.so.6, libc.so.6, libm.so.6), устанавливаются нужные значения для работы.   
Программа успешно запускается с помощью execve с аргументом data.txt

**2. Работа программы:**

Программа начинает выполняться. Происходит первый явный вызов кода из программы — открывается файл data.txt для чтения. Затем файл отображается в память с помощью mmap, что позволяет программе работать с его содержимым. После обработки данных программа открывает файл result.txt для записи результата. Результат ("Сумма чисел: 6") успешно записывается в файл result.txt с помощью системного вызова write.

**3. Завершение работы программы:**

Программа корректно завершает работу: файлы закрываются, отображённая память освобождается с помощью munmap, и программа выходит с кодом 0 (успешное завершение).

**Вывод**

В данной лабораторной работе **strace** был использован для анализа работы программы. Были проанализированы все системные вызовы, включая открытие файлов, отображение их в память, обработку данных и запись результата. Это позволило получить понимание того, как программа взаимодействует с операционной системой, и убедиться в корректной работе процесса. Вывод **strace** подтвердил, что программа успешно выполняет все необходимые операции и завершает работу без ошибок.