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: Захаров В.А.

Группа: М8О–208Б–21

Вариант: 19

Преподаватель: Соколов Андрей Алексеевич

Оценка: \_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_

Москва, 2023.

**Постановка задачи**

**Цель работы**

Приобретение практических навыков диагностики работы программного обеспечения на примере 4 лабораторной работы.

**Задание**

Провести диагностику работы 4 лабораторной работы при помощи strace, объяснить результат работы strace.

Вариант 19: Родитель перекидывает строки, введенные от пользователя свои двум детям в соотношении 80/20, те в свою очередь, выводят строки в файлы, убирая все гласные.

**Вывод strace**

viktory@DESKTOP-TU88M9R:/mnt/c/Users/User/Desktop/OS/Lab1$ strace ./a.out

execve("./a.out", ["./a.out"], 0x7ffc74589030 /\* 52 vars \*/) = 0

brk(NULL) = 0x55e222f86000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffc7e445740) = -1 EINVAL (Недопустимый аргумент)

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=100709, ...}) = 0

mmap(NULL, 100709, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f2128722000

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\360q\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\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2029224, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f2128720000

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\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68

mmap(NULL, 2036952, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f212852e000

mprotect(0x7f2128553000, 1847296, PROT\_NONE) = 0

mmap(0x7f2128553000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25000) = 0x7f2128553000

mmap(0x7f21286cb000, 303104, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19d000) = 0x7f21286cb000

mmap(0x7f2128716000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7f2128716000

mmap(0x7f212871c000, 13528, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f212871c000

close(3) = 0

arch\_prctl(ARCH\_SET\_FS, 0x7f2128721540) = 0

mprotect(0x7f2128716000, 12288, PROT\_READ) = 0

mprotect(0x55e22240a000, 4096, PROT\_READ) = 0

mprotect(0x7f2128768000, 4096, PROT\_READ) = 0

munmap(0x7f2128722000, 100709) = 0

write(0, "\320\222\320\262\320\265\320\264\320\270 \320\262 \321\201\320\273\320\265\320\264\321\203\321\216\321\211\320\270\320\271"..., 200Введи в следующий двух строках - пути к файлам, куда я буду отправлять выходные данные ^\_^

) = 200

read(0, res

"res\n", 200) = 4

read(0, res2

"res2\n", 200) = 5

openat(AT\_FDCWD, "res", O\_WRONLY|O\_CREAT|O\_APPEND, 0700) = 3

openat(AT\_FDCWD, "res2", O\_WRONLY|O\_CREAT|O\_APPEND, 0700) = 4

openat(AT\_FDCWD, "pip1", O\_RDWR|O\_CREAT, 0700) = 5

mmap(NULL, 200, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 5, 0) = 0x7f2128767000

ftruncate(5, 200) = 0

openat(AT\_FDCWD, "ex1", O\_RDWR|O\_CREAT, 0700) = 6

mmap(NULL, 1, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 6, 0) = 0x7f212873a000

ftruncate(6, 1) = 0

openat(AT\_FDCWD, "pip2", O\_RDWR|O\_CREAT, 0700) = 7

mmap(NULL, 200, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 7, 0) = 0x7f2128739000

ftruncate(7, 200) = 0

openat(AT\_FDCWD, "ex2", O\_RDWR|O\_CREAT, 0700) = 8

mmap(NULL, 8, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 8, 0) = 0x7f2128738000

ftruncate(8, 1) = 0

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f2128721810) = 36394

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f2128721810) = 36396

write(0, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \321\201\320\262\320\276\320\270 \321\201\321\202\321\200\320\276"..., 50Введите свои строки

) = 50

fstat(0, {st\_mode=S\_IFCHR|0600, st\_rdev=makedev(0x88, 0), ...}) = 0

brk(NULL) = 0x55e222f86000

brk(0x55e222fa7000) = 0x55e222fa7000

read(0, qwerty

"qwerty\n", 1024) = 7

write(0, "\320\237\320\265\321\200\320\262\321\213\320\271 \321\204\320\260\320\271\320\273\n\0\0\0\0\0\0\0\0\0\0"..., 50Первый файл

) = 50

read(0, [1] Успешно записано! [1]

ttyyu

"ttyyu\n", 1024) = 6

write(0, "\320\222\321\202\320\276\321\200\320\276\320\271 \321\204\320\260\320\271\320\273\n\0\0\0\0\0\0\0\0\0\0"..., 50Второй файл

) = 50

read(0, [2] Успешно записано! [2]

"", 1024) = 0

close(3) = 0

close(4) = 0

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=36396, si\_uid=1000, si\_status=0, si\_utime=92652, si\_stime=2} ---

unlink("ex1") = -1 ENOENT (Нет такого файла или каталога)

unlink("ex2") = -1 ENOENT (Нет такого файла или каталога)

unlink("pip1") = -1 ENOENT (Нет такого файла или каталога)

unlink("pip2") = -1 ENOENT (Нет такого файла или каталога)

exit\_group(0) = ?

+++ exited with 0 +++

**Описание работы**

execve("./a.out", ["./a.out"], 0x7ffc74589030 /\* 52 vars \*/) = 0

Исполняет программу ./a.out с ключами ["./a.out"]. Возвращает 0 – успешное выполнение.

brk(NULL) = 0x55e222f86000

Устанавливает конец сегмента данных в значение NULL, возвращает указатель на начало новой области памяти = 0x55e222f86000.

access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (Нет такого файла или каталога)

Проверяет /etc/ld.so.preload на существование (R\_OK), возвращает -1 – не существует

/etc/ld.so.preload, errno устанавливается в ENOENT (компонент пути не существует или является "висячей" символической ссылкой).

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

Открывает /etc/ld.so.cache относительно дескриптора указанного каталога - AT\_FDCWD (относительно текущего рабочего каталога вызывающего процесса) с правами доступа - O\_RDONLY|O\_CLOEXEC (на чтение и устанавливает флаг close-on-exec на новом файловом дескрипторе). Возвращает новый файловый дескриптор 3.

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=100709, ...}) = 0

Заполняет структуру указанную вторым аргументом fstat информацией об файле с файловым дескриптором 3. Возвращает 0 – успешное выполнение.

mmap(NULL, 100709, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f2128722000

Создает отображение файла с файловым дескриптором 3 в память, начиная с адресса NULL (система сама выбирает), размер = 88176 байт, с правами защиты памяти на чтение PROT\_READ, задает тип отражаемого объекта MAP\_PRIVATE - создает неразделяемое 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 = 0x7f2128722000.

close(3) = 0

Закрывает файл с файловым дескриптором 3. Возвращает 0 – успешное выполнение.

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"..., 832) = 832

Читает 832 байта данных из файла с файловым дескриптором 3 в буффер указанный вторым аргументом. Возвращает число успешно считанных байт = 832.

mprotect(0x7f2128553000, 1847296, PROT\_NONE) = 0

Контролирует доступ к области памяти начинающейся с адресса 0x7f2128553000длины 1847296 байт, доступ к памяти запрещен - PROT\_NONE. Если программой производится запрещенный этой функцией доступ к памяти, то такая программа получает сигнал SIGSEGV. Возвращает 0 – успешное завершение.

arch\_prctl(ARCH\_SET\_FS, 0x7f2128721540) = 0

Устанавливает специфичное для архитектуры состояние. Устанавливает 64 битную базу для регистра FS (ARCH\_SET\_FS) в значение 0x7f2128721540. Возвращает 0 – успешное выполнение.

ftruncate(5, 200) = 0

Изменяет длину для файла с файловым дескриптором 5 на длину = 200 байт. Если размер увеличивается новые данный заполняются нулевыми байтами, если размер уменьшается данные обрезаются. Возвращает 0 – успешное выполнение.

write(0, "\320\237\320\265\321\200\320\262\321\213\320\271 \321\204\320\260\320\271\320\273\n\0\0\0\0\0\0\0\0\0\0"..., 50Первый файл

) = 50

Выводит сообщение размером 50 байт на стандартный поток вывода сообщений «Первый файл». Возвращает количество напечатанный байт – 50.

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f2128721810) = 36394

Создает процесс-потомок с фалагами - CLONE\_CHILD\_CLEARTID (очищает id), CLONE\_CHILD\_SETTID (устанавливает id), SIGCHLD (сигнал о изменении статуса дочернего процесса), задает положение стека для процесса-потомка = NULL, задает указатель на id = 0x7f2128721810. Возвращает pid процесса-потомка.

**Вывод**

Strace оказался очень удобным инструментом, в полноценной разработке с помощью данной утилиты можно легко делать отладку и искать ошибки. Также мы можем с помощью ключа –f проследить за вызовами функций отдельных процессов, pid каждого отрисовывается перед каждым вызовом, что делает strace важным и полезным инструментом.