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: Воробьева К.Н.

Группа: М8О–201Б–21

Вариант: -

Преподаватель: Миронов Е. С.

Оценка: \_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_

Москва, 2022

**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 13: Child1 переводит строки в нижний регистр. Child2

превращает все пробельные символы в символ «\_».

**Вывод strace**

karina@MSI:~/projects/OS/build/lab4$ strace ./lab4

execve("./lab4", ["./lab4"], 0x7ffffe35d570 /\* 28 vars \*/) = 0

brk(NULL) = 0x55f35840c000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fff827ca930) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f8709feb000

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=29111, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 29111, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f8709fe3000

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\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) = 0x7f8709db9000

mmap(0x7f8709e53000, 1114112, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7f8709e53000

mmap(0x7f8709f63000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1aa000) = 0x7f8709f63000

mmap(0x7f8709fd2000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x218000) = 0x7f8709fd2000

mmap(0x7f8709fe0000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f8709fe0000

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\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) = 0x7f8709d99000

mmap(0x7f8709d9c000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f8709d9c000

mmap(0x7f8709db3000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7f8709db3000

mmap(0x7f8709db7000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7f8709db7000

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\0P\237\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 \0\0\0\5\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\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) = 0x7f8709b71000

mmap(0x7f8709b99000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7f8709b99000

mmap(0x7f8709d2e000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7f8709d2e000

mmap(0x7f8709d86000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x214000) = 0x7f8709d86000

mmap(0x7f8709d8c000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f8709d8c000

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\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) = 0x7f8709a8a000

mmap(0x7f8709a98000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7f8709a98000

mmap(0x7f8709b14000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7f8709b14000

mmap(0x7f8709b6f000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7f8709b6f000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f8709a88000

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f8709a85000

arch\_prctl(ARCH\_SET\_FS, 0x7f8709a85740) = 0

set\_tid\_address(0x7f8709a85a10) = 41

set\_robust\_list(0x7f8709a85a20, 24) = 0

rseq(0x7f8709a860e0, 0x20, 0, 0x53053053) = 0

mprotect(0x7f8709d86000, 16384, PROT\_READ) = 0

mprotect(0x7f8709b6f000, 4096, PROT\_READ) = 0

mprotect(0x7f8709db7000, 4096, PROT\_READ) = 0

mprotect(0x7f8709fd2000, 45056, PROT\_READ) = 0

mprotect(0x55f358338000, 4096, PROT\_READ) = 0

mprotect(0x7f870a025000, 8192, PROT\_READ) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

munmap(0x7f8709fe3000, 29111) = 0

getrandom("\x96\xfb\xda\x65\x1f\x9e\x81\xbf", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x55f35840c000

brk(0x55f35842d000) = 0x55f35842d000

futex(0x7f8709fe077c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

newfstatat(0, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x1), ...}, AT\_EMPTY\_PATH) = 0

read(0,

**Описание работы**

execve() – исполняет программу. Возвращает 0 – успешное выполнение.

brk(NULL) - Устанавливает конец сегмента данных в значение NULL, возвращает указатель на начало новой области памяти

access() - Проверяет на существование и на наличие прав на чтение, возвращает -1 – или не существует или нет прав на чтение, errno устанавливается в ENOENT (компонент пути не существует или является "висячей" символической ссылкой).

openat() - Открывает относительно дескриптора указанного каталога с правами доступа. Возвращает новый файловый дескриптор .

fstat() - Заполняет структуру указанную вторым аргументом fstat информацией об файле с файловым дескриптором (1-й аргумент). Возвращает 0 – успешное выполнение.

mmap() - Создает отображение файла в память. Возвращает указатель на начало отраженной памяти.

close() - Закрывает файл. Возвращает 0 – успешное выполнение.

read() - Читает данные из файла в буффер указанный вторым аргументом. Возвращает число успешно считанных байт.

mprotect() - Контролирует доступ к области памяти. Возвращает 0 – успешное завершение.

arch\_prctl() - Устанавливает специфичное для архитектуры состояние. Возвращает 0 – успешное выполнение.

munmap() - Снимает отражение из заданной области памяти. Возвращает 0 – успешное выполнение.

write() Записывает данные из буффера (второй аргумент) в файл. Возвращает число успешно записанных байт.

read() Читает данные в буффер из файл. Возвращает число успешно считанных байт.

**Вывод**

В результате данной лабораторной работы я выяснила, что при помощи strace можно анализировать работу программы, смотреть на различные системные вызовы их параметры, также можно смотреть системные вызовы по процессам, все это помогает искать неполадки в работе программы и устранять их.