Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

**Лабораторная работа №5 по курсу**

**«Операционные системы»**

Группа: М8О-209БВ-24

Студент: Тарабрина Я.Р.

Преподаватель: Миронов Е.С.

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

Дата: 10.12.25

Москва, 2025

**Постановка задачи**

Составить и отладить программу на языке Си, осуществляющую трассировку системных вызовов и анализ работы программ, разработанных в предыдущих четырёх лабораторных работах, с использованием утилиты strace (для Linux) или её аналогов.

Изучить механизм системных вызовов операционной системы, научиться анализировать взаимодействие пользовательских программ с ядром ОС, понимать, как реализованы на системном уровне функции, использованные в предыдущих работах.

**Strace 1 лабораторной**

(base) src git:(main) ✗ make run

docker run --rm -it \

--cap-add=SYS\_PTRACE --security-opt seccomp=unconfined \

-v "/Users/yana/OS/lab1/src":/lab -w /lab lab-processes:latest \

bash -c "make && strace -f -e trace=process,pipe,execve,waitpid ./parent"

make: Nothing to be done for 'all'.

**execve**("./parent", ["./parent"], 0xffffe68cd2a8 /\* 7 vars \*/) = 0

Enter filename for child1: a1.txt

Enter filename for child2: a2.txt

**clone**(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLDstrace: Process 11 attached

, child\_tidptr=0xffff99b03fd0) = 11

[pid 10] clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLDstrace: Process 12 attached

, child\_tidptr=0xffff99b03fd0) = 12

[pid 11] execve("./child", ["child", "a1.txt"], 0xfffffc577cd8 /\* 7 vars \*/Enter lines (Ctrl-D to finish):

<unfinished ...>

[pid 12] execve("./child", ["child", "a2.txt"], 0xfffffc577cd8 /\* 7 vars \*/ <unfinished ...>

[pid 11] <... execve resumed>) = 0

[pid 12] <... execve resumed>) = 0

**qwe**

**Transformed: ewq**

**ewq**

**Transformed: qwe**

[pid 10] wait4(11, <unfinished ...>

[pid 11] exit\_group(0 <unfinished ...>

[pid 12] exit\_group(0 <unfinished ...>

[pid 11] <... exit\_group resumed>) = ?

[pid 12] <... exit\_group resumed>) = ?

[pid 11] +++ exited with 0 +++

[pid 10] <... wait4 resumed>[{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 11

[pid 12] +++ exited with 0 +++

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=11, si\_uid=0, si\_status=0, si\_utime=0, si\_stime=0} ---

**wait4**(12, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 12

**exit\_group(0)** = ?

+++ exited with 0 +++

Анализ системных вызовов, полученный с помощью утилиты `strace`, демонстрирует корректную реализацию взаимодействия между родительским и дочерними процессами в рамках выполнения программы. Изначально запускается исполняемый файл `./parent`, что подтверждается успешным вызовом `execve`, после чего программа запрашивает у пользователя имена файлов для двух дочерних процессов — `a1.txt` и `a2.txt`. Далее родительский процесс (с PID 10) создаёт два дочерних процесса последовательными вызовами `clone` с флагами, характерными для стандартного создания процессов в Linux-системах через `fork`-подобное поведение. В результате создаются процессы с идентификаторами 11 и 12, каждый из которых немедленно выполняет `execve`, заменяя свой адресное пространство на образ программы `./child` с соответствующим аргументом командной строки — `a1.txt` или `a2.txt`.

Оба дочерних процесса запускаются и начинают выполнять свою логику: они ожидают ввода строк от пользователя, после чего каждая строка инвертируется (например, ввод `qwe` преобразуется в `ewq`) и выводится в стандартный поток. Это свидетельствует о том, что `child` корректно обрабатывает данные и взаимодействует со стандартным вводом и выводом. При этом процессы работают параллельно, что подтверждается переплетением их системных вызовов в логе `strace`. После завершения пользовательского ввода (посылка EOF через Ctrl-D), оба дочерних процесса завершаются с помощью системного вызова `exit\_group(0)`, что указывает на штатное завершение без ошибок.

Родительский процесс, в свою очередь, ожидает завершения каждого из дочерних процессов с помощью вызова `wait4`. Сначала он дожидается процесса с PID 11, получает корректное подтверждение его завершения (`WIFEXITED(s) && WEXITSTATUS(s) == 0`), а затем — процесса с PID 12. Между этими вызовами система генерирует сигнал `SIGCHLD`, уведомляя родительский процесс о завершении одного из потомков, что является стандартным механизмом взаимодействия ядра с пользовательскими процессами при управлении деревом процессов. После успешного ожидания обоих дочерних процессов родитель сам завершается через `exit\_group(0)`, завершая тем самым выполнение всей программы.

Таким образом, лог `strace` подтверждает, что реализованная схема межпроцессного взаимодействия работает корректно: процессы создаются, выполняют свою задачу параллельно, завершаются штатно, а родительский процесс корректно дожидается их окончания, предотвращая появление "зомби"-процессов и обеспечивая упорядоченное завершение всей системы процессов.

**Strace 2 лабораторной**

lab2 docker run convolution-lab strace -f ./convolution 4 100 3 1 2>&1

execve("./convolution", ["./convolution", "4", "100", "3", "1"], 0xfffff39f6c38 /\* 3 vars \*/) = 0

brk(NULL) = 0xaaaaf54f3000

faccessat(AT\_FDCWD, "/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=8663, ...}) = 0

mmap(NULL, 8663, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0xffffb9e2e000

close(3) = 0

openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libpthread.so.0", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0Ha\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=164304, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffffb9e2c000

mmap(NULL, 197624, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0xffffb9dd2000

mprotect(0xffffb9dee000, 61440, PROT\_NONE) = 0

mmap(0xffffb9dfd000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1b000) = 0xffffb9dfd000

mmap(0xffffb9dff000, 13304, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0xffffb9dff000

close(3) = 0

openat(AT\_FDCWD, "/lib/aarch64-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\267\0\1\0\0\0`\17\2\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=1450832, ...}) = 0

mmap(NULL, 1519552, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0xffffb9c5f000

mprotect(0xffffb9dba000, 61440, PROT\_NONE) = 0

mmap(0xffffb9dc9000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x15a000) = 0xffffb9dc9000

mmap(0xffffb9dcf000, 12224, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0xffffb9dcf000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffffb9e2a000

mprotect(0xffffb9dc9000, 16384, PROT\_READ) = 0

mprotect(0xffffb9dfd000, 4096, PROT\_READ) = 0

mprotect(0xaaaac5952000, 4096, PROT\_READ) = 0

mprotect(0xffffb9e34000, 4096, PROT\_READ) = 0

munmap(0xffffb9e2e000, 8663) = 0

set\_tid\_address(0xffffb9e2a0e0) = 9

set\_robust\_list(0xffffb9e2a0f0, 24) = 0

rt\_sigaction(SIGRTMIN, {sa\_handler=0xffffb9dd7bd0, sa\_mask=[], sa\_flags=SA\_SIGINFO}, NULL, 8) = 0

rt\_sigaction(SIGRT\_1, {sa\_handler=0xffffb9dd7c90, sa\_mask=[], sa\_flags=SA\_RESTART|SA\_SIGINFO}, NULL, 8) = 0

rt\_sigprocmask(SIG\_UNBLOCK, [RTMIN RT\_1], NULL, 8) = 0

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

fstat(1, {st\_mode=S\_IFIFO|0600, st\_size=0, ...}) = 0

brk(NULL) = 0xaaaaf54f3000

brk(0xaaaaf5514000) = 0xaaaaf5514000

brk(0xaaaaf5535000) = 0xaaaaf5535000

clock\_gettime(CLOCK\_PROCESS\_CPUTIME\_ID, {tv\_sec=0, tv\_nsec=1695625}) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0xffffb945e000

mprotect(0xffffb945f000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0xffffb9c5dae0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTIDstrace: Process 10 attached

, parent\_tid=[10], tls=0xffffb9c5e8e0, child\_tidptr=0xffffb9c5e2b0) = 10

[pid 9] mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0xffffb8c5d000

[pid 10] set\_robust\_list(0xffffb9c5e2c0, 24 <unfinished ...>

[pid 9] mprotect(0xffffb8c5e000, 8388608, PROT\_READ|PROT\_WRITE <unfinished ...>

[pid 10] <... set\_robust\_list resumed>) = 0

[pid 10] mmap(NULL, 134217728, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_NORESERVE, -1, 0 <unfinished ...>

[pid 9] <... mprotect resumed>) = 0

[pid 10] <... mmap resumed>) = 0xffffb0c00000

[pid 10] munmap(0xffffb0c00000, 54525952) = 0

[pid 10] munmap(0xffffb8000000, 12582912 <unfinished ...>

[pid 9]

clone(child\_stack=0xffffb945cae0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTIDstrace: Process 11 attached

<unfinished ...>

[pid 10] <... munmap resumed>) = 0

[pid 9] <... clone resumed>, parent\_tid=[11], tls=0xffffb945d8e0, child\_tidptr=0xffffb945d2b0) = 11

[pid 11] set\_robust\_list(0xffffb945d2c0, 24 <unfinished ...>

[pid 10] mprotect(0xffffb4000000, 135168, PROT\_READ|PROT\_WRITE <unfinished ...>

[pid 9] mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0 <unfinished ...>

[pid 11] <... set\_robust\_list resumed>) = 0

[pid 9] <... mmap resumed>) = 0xffffb845c000

[pid 11] mmap(NULL, 134217728, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_NORESERVE, -1, 0 <unfinished ...>

[pid 10] <... mprotect resumed>) = 0

[pid 9] mprotect(0xffffb845d000, 8388608, PROT\_READ|PROT\_WRITE <unfinished ...>

[pid 11] <... mmap resumed>) = 0xffffac000000

[pid 9] <... mprotect resumed>) = 0

[pid 11] munmap(0xffffb0000000, 67108864 <unfinished ...>

[pid 10] openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

[pid 10] fstat(3, {st\_mode=S\_IFREG|0644, st\_size=8663, ...}) = 0

[pid 10] mmap(NULL, 8663, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0xffffb9e2e000

[pid 10] close(3) = 0

[pid 10] openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 9]

clone(child\_stack=0xffffb8c5bae0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID <unfinished ...>

[pid 11] <... munmap resumed>) = 0

[pid 10] <... openat resumed>) = 3

[pid 11] mprotect(0xffffac000000, 135168, PROT\_READ|PROT\_WRITE <unfinished ...>

[pid 10] read(3, <unfinished ...>

[pid 11] <... mprotect resumed>) = 0

strace: Process 12 attached

[pid 11] futex(0xffffb9e359e8, \_WAIT\_PRIVATE, 2, NULL <unfinished ...>

[pid 10] <... read resumed>"\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\320)\0\0\0\0\0\0"..., 832) = 832

[pid 12] set\_robust\_list(0xffffb8c5c2c0, 24) = 0

[pid 10] fstat(3, <unfinished ...>

[pid 9] <... clone resumed>, parent\_tid=[12], tls=0xffffb8c5c8e0, child\_tidptr=0xffffb8c5c2b0) = 12

[pid 12] mmap(0xffffb0000000, 67108864, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_NORESERVE, -1, 0 <unfinished ...>

[pid 10] <... fstat resumed>{st\_mode=S\_IFREG|0644, st\_size=80200, ...}) = 0

[pid 9] mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0 <unfinished ...>

[pid 12] <... mmap resumed>) = 0xffffa8000000

[pid 10] mmap(NULL, 144472, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0xffffb8438000

[pid 10] mprotect(0xffffb844b000, 61440, PROT\_NONE) = 0

[pid 10] mmap(0xffffb845a000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x12000) = 0xffffb845a000

[pid 10] close(3) = 0

[pid 10] mprotect(0xffffb845a000, 4096, PROT\_READ) = 0

[pid 10] munmap(0xffffb9e2e000, 8663) = 0

[pid 10] futex(0xffffb9e359e8, FUTEX\_WAKE\_PRIVATE, 1) = 1

[pid 10] futex(0xffffb845b234, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

[pid 10] madvise(0xffffb945e000, 8253440, MADV\_DONTNEED) = 0

[pid 10] exit(0) = ?

[pid 10] +++ exited with 0 +++

[pid 11] <... futex resumed>) = 0

[pid 9] <... mmap resumed>) = 0xffffb37ff000

[pid 9] mprotect(0xffffb3800000, 8388608, PROT\_READ|PROT\_WRITE <unfinished ...>

[pid 11] futex(0xffffb9e359e8, FUTEX\_WAKE\_PRIVATE, 1 <unfinished ...>

[pid 9] <... mprotect resumed>) = 0

[pid 11] <... futex resumed>) = 0

[pid 9]

clone(child\_stack=0xffffb3ffeae0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID <unfinished ...>

[pid 12] mprotect(0xffffa8000000, 135168, PROT\_READ|PROT\_WRITE <unfinished ...>

[pid 11] madvise(0xffffb8c5d000, 8253440, MADV\_DONTNEED <unfinished ...>

[pid 12] <... mprotect resumed>) = 0

strace: Process 13 attached

[pid 12] madvise(0xffffb845c000, 8253440, MADV\_DONTNEED) = 0

[pid 12] exit(0 <unfinished ...>

[pid 9] <... clone resumed>, parent\_tid=[13], tls=0xffffb3fff8e0, child\_tidptr=0xffffb3fff2b0) = 13

[pid 12] <... exit resumed>) = ?

[pid 11] <... madvise resumed>) = 0

[pid 13] set\_robust\_list(0xffffb3fff2c0, 24 <unfinished ...>

[pid 9] futex(0xffffb945d2b0, FUTEX\_WAIT, 11, NULL <unfinished ...>

[pid 13] <... set\_robust\_list resumed>) = 0

[pid 12] +++ exited with 0 +++

[pid 11] exit(0) = ?

[pid 13] madvise(0xffffb37ff000, 8253440, MADV\_DONTNEED <unfinished ...>

[pid 9] <... futex resumed>) = 0

[pid 13] <... madvise resumed>) = 0

[pid 11] +++ exited with 0 +++

[pid 9] futex(0xffffb3fff2b0, FUTEX\_WAIT, 13, NULL <unfinished ...>

[pid 13] exit(0) = ?

[pid 13] +++ exited with 0 +++

<... futex resumed>) = 0

clock\_gettime(CLOCK\_PROCESS\_CPUTIME\_ID, {tv\_sec=0, tv\_nsec=4201879}) = 0

write(1, "\n=== \320\234\320\235\320\236\320\223\320\236\320\237\320\236\320\242\320\236\320\247\320\235\320\220\320\257 "..., 852) = 852

exit\_group(0) = ?

Анализ системного трассирования выполнения программы `./convolution` с параметрами `4 100 3 1` показывает корректную инициализацию многопоточного приложения, основанного на использовании библиотеки `pthread`, и его штатное завершение. При запуске процесс загружает необходимые разделяемые библиотеки — `libpthread.so.0` и `libc.so.6` — через стандартные вызовы `openat`, `mmap` и `mprotect`, что типично для динамически линкованных программ в Linux-окружении. После завершения этапа загрузки и инициализации памяти основной поток (с PID 9) приступает к созданию дополнительных потоков выполнения с помощью системного вызова `clone` с флагами, характерными для POSIX-потоков: `CLONE\_THREAD`, `CLONE\_VM`, `CLONE\_SIGHAND` и другими, что обеспечивает совместное использование адресного пространства, файловых дескрипторов и таблицы сигналов между потоками.

В ходе выполнения создаются три дополнительных потока (PID 10, 11 и 12), каждый из которых проходит через фазу инициализации: выделяется стек (`mmap` с флагом `MAP\_STACK`), настраивается TLS (Thread Local Storage), регистрируется robust futex list для корректной обработки блокировок при аварийном завершении потока. Один из потоков (PID 10) дополнительно загружает `libgcc\_s.so.1`, что может быть связано с использованием операций с плавающей точкой или специфических арифметических инструкций, требующих поддержки со стороны компилятора. После выполнения своей вычислительной задачи каждый из рабочих потоков завершается с помощью вызова `exit(0)`, при этом ядро корректно освобождает ресурсы, включая отображённые в память регионы, с использованием системного вызова `madvise(..., MADV\_DONTNEED)`, который сигнализирует о том, что содержимое страниц больше не требуется и может быть отброшено.

Основной поток (PID 9) ожидает завершения всех порождённых потоков с помощью блокирующих вызовов `futex(..., FUTEX\_WAIT, ...)`, что является стандартным механизмом синхронизации в библиотеке `pthread`. После того как все дочерние потоки завершили свою работу, основной поток фиксирует общее время выполнения с помощью `clock\_gettime(CLOCK\_PROCESS\_CPUTIME\_ID, ...)`, выводит результат своей работы в стандартный поток (`write(1, "...", 852)`), содержащий, судя по длине и кодировке (UTF-8), подробную информацию о ходе выполнения или итоговую статистику многопоточной свёртки, и завершает программу вызовом `exit\_group(0)`, который обеспечивает одновременное завершение всех потоков процесса.

Весь процесс выполнения демонстрирует корректное использование многопоточности: потоки создаются, работают параллельно, не нарушая целостности адресного пространства, синхронизируются через стандартные примитивы ядра (`futex`), и завершаются упорядоченно. Отсутствие ошибок в системных вызовах (все возвращаемые значения соответствуют успешному выполнению) и код завершения `0` подтверждают стабильную и корректную работу программы свёртки, реализованной с применением многопоточности. Таким образом, лог `strace` свидетельствует о том, что архитектура приложения соответствует ожидаемой модели параллельных вычислений, а взаимодействие с операционной системой осуществляется в рамках стандартных POSIX-интерфейсов.

**Strace 3 лабораторной**

root@5630c98cc6e4:/lab# strace -f ./parent

execve("./parent", ["./parent"], 0xffffc741cc98 /\* 7 vars \*/) = 0

brk(NULL) = 0xaaaae6b9f000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffff8f7a0000

faccessat(AT\_FDCWD, "/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=6211, ...}) = 0

mmap(NULL, 6211, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0xffff8f79e000

close(3) = 0

openat(AT\_FDCWD, "/lib/aarch64-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\267\0\1\0\0\0\300$\2\0\0\0\0\0"..., 832) = 832

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=1782152, ...}) = 0

mmap(NULL, 1892400, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_DENYWRITE, -1, 0) = 0xffff8f598000

mmap(0xffff8f5a0000, 1826864, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff8f5a0000

munmap(0xffff8f598000, 32768) = 0

munmap(0xffff8f75f000, 28720) = 0

mprotect(0xffff8f73e000, 61440, PROT\_NONE) = 0

mmap(0xffff8f74d000, 20480, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ad000) = 0xffff8f74d000

mmap(0xffff8f752000, 49200, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0xffff8f752000

close(3) = 0

set\_tid\_address(0xffff8f7a0fd0) = 12

set\_robust\_list(0xffff8f7a0fe0, 24) = 0

rseq(0xffff8f7a16c0, 0x20, 0, 0xd428bc00) = 0

mprotect(0xffff8f74d000, 12288, PROT\_READ) = 0

mprotect(0xaaaada20f000, 4096, PROT\_READ) = 0

mprotect(0xffff8f7a5000, 8192, PROT\_READ) = 0

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

munmap(0xffff8f79e000, 6211) = 0

openat(AT\_FDCWD, "shared.dat", O\_RDWR|O\_CREAT, 0666) = 3

ftruncate(3, 1028) = 0

mmap(NULL, 1028, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0xffff8f79f000

close(3) = 0

rt\_sigaction(SIGUSR2, {sa\_handler=0xaaaada1f1040, sa\_mask=[], sa\_flags=0}, NULL, 8) = 0

rt\_sigprocmask(SIG\_BLOCK, ~[], [], 8) = 0

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLDstrace: Process 13 attached

, child\_tidptr=0xffff8f7a0fd0) = 13

[pid 13] set\_robust\_list(0xffff8f7a0fe0, 24 <unfinished ...>

[pid 12] rt\_sigprocmask(SIG\_SETMASK, [], NULL, 8) = 0

[pid 13] <... set\_robust\_list resumed>) = 0

[pid 12] rt\_sigprocmask(SIG\_BLOCK, ~[], <unfinished ...>

[pid 13] rt\_sigprocmask(SIG\_SETMASK, [], <unfinished ...>

[pid 12] <... rt\_sigprocmask resumed>[], 8) = 0

[pid 13] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 12] clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLDstrace: Process 14 attached

<unfinished ...>

[pid 13] execve("./child", ["child", "shared.dat"], 0xffffdd4c5a68 /\* 7 vars \*/ <unfinished ...>

[pid 12] <... clone resumed>, child\_tidptr=0xffff8f7a0fd0) = 14

[pid 14] set\_robust\_list(0xffff8f7a0fe0, 24 <unfinished ...>

[pid 12] rt\_sigprocmask(SIG\_SETMASK, [], <unfinished ...>

[pid 14] <... set\_robust\_list resumed>) = 0

[pid 12] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 14] rt\_sigprocmask(SIG\_SETMASK, [], <unfinished ...>

[pid 13] <... execve resumed>) = 0

[pid 12] fstat(1, <unfinished ...>

[pid 14] <... rt\_sigprocmask resumed>NULL, 8) = 0

[pid 13] brk(NULL <unfinished ...>

[pid 12] <... fstat resumed>{st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}) = 0

[pid 14] execve("./child", ["child", "shared.dat"], 0xffffdd4c5a68 /\* 7 vars \*/ <unfinished ...>

[pid 13] <... brk resumed>) = 0xaaaae8511000

[pid 13] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 12] getrandom("\x9e\x52\xea\x04\x94\xda\xf1\xcc", 8, GRND\_NONBLOCK) = 8

[pid 13] <... mmap resumed>) = 0xffffa4684000

[pid 12] brk(NULL <unfinished ...>

[pid 13] faccessat(AT\_FDCWD, "/etc/ld.so.preload", R\_OK <unfinished ...>

[pid 12] <... brk resumed>) = 0xaaaae6b9f000

[pid 14] <... execve resumed>) = 0

[pid 13] <... faccessat resumed>) = -1 ENOENT (No such file or directory)

[pid 12] brk(0xaaaae6bc0000 <unfinished ...>

[pid 14] brk(NULL <unfinished ...>

[pid 13] openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 12] <... brk resumed>) = 0xaaaae6bc0000

[pid 14] <... brk resumed>) = 0xaaaae9e8b000

[pid 12] write(1, "Enter lines (Ctrl-D to finish):\n", 32 <unfinished ...>

[pid 14] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 13] <... openat resumed>) = 3

[pid 14] <... mmap resumed>) = 0xffff8592c000

Enter lines (Ctrl-D to finish):

[pid 13] fstat(3, <unfinished ...>

[pid 12] <... write resumed>) = 32

[pid 12] fstat(0, <unfinished ...>

[pid 14] faccessat(AT\_FDCWD, "/etc/ld.so.preload", R\_OK <unfinished ...>

[pid 13] <... fstat resumed>{st\_mode=S\_IFREG|0644, st\_size=6211, ...}) = 0

[pid 12] <... fstat resumed>{st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}) = 0

[pid 14] <... faccessat resumed>) = -1 ENOENT (No such file or directory)

[pid 13] mmap(NULL, 6211, PROT\_READ, MAP\_PRIVATE, 3, 0 <unfinished ...>

[pid 14] openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 12] read(0, <unfinished ...>

[pid 14] <... openat resumed>) = 3

[pid 13] <... mmap resumed>) = 0xffffa4682000

[pid 14] fstat(3, <unfinished ...>

[pid 13] close(3 <unfinished ...>

[pid 14] <... fstat resumed>{st\_mode=S\_IFREG|0644, st\_size=6211, ...}) = 0

[pid 13] <... close resumed>) = 0

[pid 14] mmap(NULL, 6211, PROT\_READ, MAP\_PRIVATE, 3, 0 <unfinished ...>

[pid 13] openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 14] <... mmap resumed>) = 0xffff8592a000

[pid 14] close(3 <unfinished ...>

[pid 13] <... openat resumed>) = 3

[pid 14] <... close resumed>) = 0

[pid 13] read(3, <unfinished ...>

[pid 14] openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 13] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\300$\2\0\0\0\0\0"..., 832) = 832

[pid 13] fstat(3, <unfinished ...>

[pid 14] <... openat resumed>) = 3

[pid 14] read(3, <unfinished ...>

[pid 13] <... fstat resumed>{st\_mode=S\_IFREG|0755, st\_size=1782152, ...}) = 0

[pid 14] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\300$\2\0\0\0\0\0"..., 832) = 832

[pid 14] fstat(3, <unfinished ...>

[pid 13] mmap(NULL, 1892400, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_DENYWRITE, -1, 0 <unfinished ...>

[pid 14] <... fstat resumed>{st\_mode=S\_IFREG|0755, st\_size=1782152, ...}) = 0

[pid 13] <... mmap resumed>) = 0xffffa447c000

[pid 14] mmap(NULL, 1892400, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_DENYWRITE, -1, 0 <unfinished ...>

[pid 13] mmap(0xffffa4480000, 1826864, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0 <unfinished ...>

[pid 14] <... mmap resumed>) = 0xffff85724000

[pid 13] <... mmap resumed>) = 0xffffa4480000

[pid 14] mmap(0xffff85730000, 1826864, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffff85730000

[pid 14] munmap(0xffff85724000, 49152 <unfinished ...>

[pid 13] munmap(0xffffa447c000, 16384 <unfinished ...>

[pid 14] <... munmap resumed>) = 0

[pid 13] <... munmap resumed>) = 0

[pid 14] munmap(0xffff858ef000, 12336 <unfinished ...>

[pid 13] munmap(0xffffa463f000, 45104 <unfinished ...>

[pid 14] <... munmap resumed>) = 0

[pid 13] <... munmap resumed>) = 0

[pid 14] mprotect(0xffff858ce000, 61440, PROT\_NONE <unfinished ...>

[pid 13] mprotect(0xffffa461e000, 61440, PROT\_NONE <unfinished ...>

[pid 14] <... mprotect resumed>) = 0

[pid 13] <... mprotect resumed>) = 0

[pid 14] mmap(0xffff858dd000, 20480, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ad000) = 0xffff858dd000

[pid 14] mmap(0xffff858e2000, 49200, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 13] mmap(0xffffa462d000, 20480, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ad000 <unfinished ...>

[pid 14] <... mmap resumed>) = 0xffff858e2000

[pid 14] close(3 <unfinished ...>

[pid 13] <... mmap resumed>) = 0xffffa462d000

[pid 14] <... close resumed>) = 0

[pid 13] mmap(0xffffa4632000, 49200, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 14] set\_tid\_address(0xffff8592cfd0 <unfinished ...>

[pid 13] <... mmap resumed>) = 0xffffa4632000

[pid 14] <... set\_tid\_address resumed>) = 14

[pid 13] close(3 <unfinished ...>

[pid 14] set\_robust\_list(0xffff8592cfe0, 24 <unfinished ...>

[pid 13] <... close resumed>) = 0

[pid 14] <... set\_robust\_list resumed>) = 0

[pid 13] set\_tid\_address(0xffffa4684fd0 <unfinished ...>

[pid 14] rseq(0xffff8592d6c0, 0x20, 0, 0xd428bc00 <unfinished ...>

[pid 13] <... set\_tid\_address resumed>) = 13

[pid 14] <... rseq resumed>) = 0

[pid 13] set\_robust\_list(0xffffa4684fe0, 24) = 0

[pid 14] mprotect(0xffff858dd000, 12288, PROT\_READ <unfinished ...>

[pid 13] rseq(0xffffa46856c0, 0x20, 0, 0xd428bc00 <unfinished ...>

[pid 14] <... mprotect resumed>) = 0

[pid 13] <... rseq resumed>) = 0

[pid 14] mprotect(0xaaaac739f000, 4096, PROT\_READ <unfinished ...>

[pid 13] mprotect(0xffffa462d000, 12288, PROT\_READ <unfinished ...>

[pid 14] <... mprotect resumed>) = 0

[pid 14] mprotect(0xffff85931000, 8192, PROT\_READ <unfinished ...>

[pid 13] <... mprotect resumed>) = 0

[pid 14] <... mprotect resumed>) = 0

[pid 13] mprotect(0xaaaaae87f000, 4096, PROT\_READ <unfinished ...>

[pid 14] prlimit64(0, RLIMIT\_STACK, NULL, <unfinished ...>

[pid 13] <... mprotect resumed>) = 0

[pid 14] <... prlimit64 resumed>{rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

[pid 13] mprotect(0xffffa4689000, 8192, PROT\_READ <unfinished ...>

[pid 14] munmap(0xffff8592a000, 6211 <unfinished ...>

[pid 13] <... mprotect resumed>) = 0

[pid 14] <... munmap resumed>) = 0

[pid 13] prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

[pid 14] openat(AT\_FDCWD, "shared.dat", O\_RDWR <unfinished ...>

[pid 13] munmap(0xffffa4682000, 6211 <unfinished ...>

[pid 14] <... openat resumed>) = 3

[pid 13] <... munmap resumed>) = 0

[pid 14] mmap(NULL, 1028, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0xffff8592b000

[pid 13] openat(AT\_FDCWD, "shared.dat", O\_RDWR <unfinished ...>

[pid 14] close(3 <unfinished ...>

[pid 13] <... openat resumed>) = 3

[pid 14] <... close resumed>) = 0

[pid 13] mmap(NULL, 1028, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0 <unfinished ...>

[pid 14] rt\_sigaction(SIGUSR1, {sa\_handler=0xaaaac7380c80, sa\_mask=[], sa\_flags=0}, <unfinished ...>

[pid 13] <... mmap resumed>) = 0xffffa4683000

[pid 14] <... rt\_sigaction resumed>NULL, 8) = 0

[pid 13] close(3 <unfinished ...>

[pid 14] getppid( <unfinished ...>

[pid 13] <... close resumed>) = 0

[pid 14] <... getppid resumed>) = 12

[pid 13] rt\_sigaction(SIGUSR1, {sa\_handler=0xaaaaae860c80, sa\_mask=[], sa\_flags=0}, <unfinished ...>

[pid 14] ppoll(NULL, 0, NULL, NULL, 0 <unfinished ...>

[pid 13] <... rt\_sigaction resumed>NULL, 8) = 0

[pid 13] getppid() = 12

[pid 13] ppoll(NULL, 0, NULL, NULL, 0^C <unfinished ...>

[pid 12] <... read resumed>0xaaaae6b9f730, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

strace: Process 12 detached

strace: Process 13 detached

strace: Process 14 detached

root@5630c98cc6e4:/lab#

exit

Анализ системного трассирования выполнения программы `./parent` с помощью утилиты `strace` демонстрирует корректную реализацию межпроцессного взаимодействия на основе разделяемой памяти и сигналов. В начале родительский процесс (PID 12) инициализирует общую область памяти, создавая файл `shared.dat` с флагами `O\_RDWR | O\_CREAT` и устанавливая его размер в 1028 байт с помощью `ftruncate`. Эта область затем отображается в адресное пространство процесса через `mmap` с флагом `MAP\_SHARED`, что позволяет другим процессам, имеющим доступ к этому файлу, разделять одну и ту же память. После настройки общей памяти родительский процесс регистрирует обработчик сигнала `SIGUSR2`, который, судя по логике, будет использоваться для синхронизации с дочерними процессами.

Далее родитель последовательно создаёт два дочерних процесса с помощью системного вызова `clone` с флагами, характерными для классического `fork`-подобного поведения (`CLONE\_CHILD\_CLEARTID`, `CLONE\_CHILD\_SETTID`, `SIGCHLD`). После создания каждый дочерний процесс (PID 13 и 14) немедленно выполняет `execve`, заменяя свой образ на программу `./child` с аргументом `shared.dat`, что указывает на их предназначение — работа с общей памятью. Оба дочерних процесса проходят полную стадию динамической загрузки: они загружают `libc.so.6`, инициализируют TLS и robust futex list, а затем самостоятельно открывают файл `shared.dat` и отображают его в своё адресное пространство с теми же параметрами (`PROT\_READ|PROT\_WRITE`, `MAP\_SHARED`), обеспечивая тем самым доступ к той же области памяти, что и у родителя.

После инициализации оба дочерних процесса устанавливают обработчики сигнала `SIGUSR1` и вызывают `getppid()`, подтверждая, что их родителем является процесс с PID 12. Затем они переходят в режим ожидания с помощью `ppoll(NULL, 0, NULL, NULL, 0)`, что эквивалентно бесконечному ожиданию сигналов, не потребляя процессорного времени. Это указывает на то, что дальнейшее взаимодействие между процессами предполагается посредством асинхронных сигналов — вероятно, родительский процесс должен был отправить `SIGUSR1` каждому из дочерних, чтобы инициировать обработку данных в общей памяти.

Одновременно с этим родительский процесс выводит приглашение `Enter lines (Ctrl-D to finish):` и вызывает `read(0, ...)`, ожидая ввода от пользователя. Однако в этот момент пользователь прерывает выполнение программы с помощью комбинации `Ctrl-C`, что приводит к генерации сигнала `SIGINT`. В ответ на это системный вызов `read` прерывается с ошибкой `ERESTARTSYS`, и все процессы — родительский и оба дочерних — завершаются, о чём свидетельствует отсоединение (`detached`) всех трёх PID в выводе `strace`.

Таким образом, лог `strace` подтверждает корректную настройку разделяемой памяти, успешное создание и инициализацию дочерних процессов, а также ожидаемую модель взаимодействия на основе сигналов и общей памяти. Однако выполнение программы было прервано до завершения основной логики работы, поэтому финальный обмен данными и синхронизация не были реализованы. Тем не менее, сама архитектура взаимодействия между процессами реализована в соответствии с принципами POSIX и демонстрирует корректное использование системных вызовов `mmap`, `clone`, `execve` и управления сигналами.системах.

**Strace 4 лабораторной**

9 execve("./main\_runtime", ["./main\_runtime"], 0xffffd90c0e78 /\* 4 vars \*/) = 0

9 brk(NULL) = 0xaaab02d7a000

9 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffffa77ea000

9 faccessat(AT\_FDCWD, "/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

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

9 newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=9768, ...}, AT\_EMPTY\_PATH) = 0

9 mmap(NULL, 9768, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0xffffa77e7000

9 close(3) = 0

9 openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = 3

9 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

9 newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=2198944, ...}, AT\_EMPTY\_PATH) = 0

9 mmap(NULL, 2340896, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffffa757a000

9 mmap(0xffffa7580000, 2275360, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffffa7580000

9 munmap(0xffffa757a000, 24576) = 0

9 munmap(0xffffa77ac000, 38944) = 0

9 mprotect(0xffffa778b000, 65536, PROT\_NONE) = 0

9 mmap(0xffffa779b000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x20b000) = 0xffffa779b000

9 mmap(0xffffa77a9000, 10272, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0xffffa77a9000

9 close(3) = 0

9 openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3

9 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

9 newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=84296, ...}, AT\_EMPTY\_PATH) = 0

9 mmap(NULL, 213704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffffa754b000

9 mmap(0xffffa7550000, 148168, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffffa7550000

9 munmap(0xffffa754b000, 20480) = 0

9 munmap(0xffffa7575000, 41672) = 0

9 mprotect(0xffffa7564000, 61440, PROT\_NONE) = 0

9 mmap(0xffffa7573000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x13000) = 0xffffa7573000

9 close(3) = 0

9 openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

9 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\360u\2\0\0\0\0\0"..., 832) = 832

9 newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=1661968, ...}, AT\_EMPTY\_PATH) = 0

9 mmap(NULL, 1830504, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffffa7391000

9 mmap(0xffffa73a0000, 1764968, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffffa73a0000

9 munmap(0xffffa7391000, 61440) = 0

9 munmap(0xffffa754f000, 3688) = 0

9 mprotect(0xffffa752e000, 61440, PROT\_NONE) = 0

9 mmap(0xffffa753d000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x18d000) = 0xffffa753d000

9 mmap(0xffffa7543000, 48744, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0xffffa7543000

9 close(3) = 0

9 openat(AT\_FDCWD, "/lib/aarch64-linux-gnu/libm.so.6", O\_RDONLY|O\_CLOEXEC) = 3

9 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

9 newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=551064, ...}, AT\_EMPTY\_PATH) = 0

9 mmap(NULL, 680048, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffffa72f9000

9 mmap(0xffffa7300000, 614512, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffffa7300000

9 munmap(0xffffa72f9000, 28672) = 0

9 munmap(0xffffa7397000, 32880) = 0

9 mprotect(0xffffa7386000, 61440, PROT\_NONE) = 0

9 mmap(0xffffa7395000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x85000) = 0xffffa7395000

9 close(3) = 0

9 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffffa77e5000

9 set\_tid\_address(0xffffa77e5af0) = 9

9 set\_robust\_list(0xffffa77e5b00, 24) = 0

9 rseq(0xffffa77e61c0, 0x20, 0, 0xd428bc00) = 0

9 mprotect(0xffffa753d000, 16384, PROT\_READ) = 0

9 mprotect(0xffffa7395000, 4096, PROT\_READ) = 0

9 mprotect(0xffffa7573000, 4096, PROT\_READ) = 0

9 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffffa77e3000

9 mprotect(0xffffa779b000, 45056, PROT\_READ) = 0

9 mprotect(0xaaaacfb22000, 4096, PROT\_READ) = 0

9 mprotect(0xffffa77f0000, 8192, PROT\_READ) = 0

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

9 munmap(0xffffa77e7000, 9768) = 0

9 getrandom("\x3d\xbf\xbc\x33\xd0\xc3\x0d\x9f", 8, GRND\_NONBLOCK) = 8

9 brk(NULL) = 0xaaab02d7a000

9 brk(0xaaab02d9b000) = 0xaaab02d9b000

9 futex(0xffffa77a97a4, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

9 newfstatat(AT\_FDCWD, "libs/libcalc1.so", {st\_mode=S\_IFREG|0755, st\_size=8216, ...}, 0) = 0

9 openat(AT\_FDCWD, "libs/libcalc1.so", O\_RDONLY|O\_CLOEXEC) = 3

9 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0\267\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

9 newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=8216, ...}, AT\_EMPTY\_PATH) = 0

9 getcwd("/work", 128) = 6

9 mmap(NULL, 135216, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0xffffa72de000

9 mmap(0xffffa72e0000, 69680, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffffa72e0000

9 munmap(0xffffa72de000, 8192) = 0

9 munmap(0xffffa72f2000, 53296) = 0

9 mprotect(0xffffa72e1000, 61440, PROT\_NONE) = 0

9 mmap(0xffffa72f0000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0) = 0xffffa72f0000

9 close(3) = 0

9 mprotect(0xffffa72f0000, 4096, PROT\_READ) = 0

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

9 read(0, "2 12 3\n", 1024) = 7

9 newfstatat(1, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0

9 write(1, "2.61304\n", 8) = 8

9 read(0, "", 1024) = 0

9 munmap(0xffffa72e0000, 69680) = 0

9 exit\_group(0) = ?

9 +++ exited with 0 +++

Анализ системного трассирования выполнения программы `./main\_runtime` показывает корректную работу приложения, реализующего динамическую загрузку пользовательской разделяемой библиотеки и обработку пользовательского ввода. В начале выполнения процесс (PID 9) проходит стандартную стадию инициализации: загружает необходимые системные библиотеки — `libstdc++.so.6`, `libgcc\_s.so.1`, `libc.so.6` и `libm.so.6` — через последовательные вызовы `openat`, `mmap` и `mprotect`, что характерно для динамически линкованных C++-приложений в Linux. После завершения этапа загрузки и настройки памяти, включая инициализацию TLS и robust futex list, программа переходит к выполнению своей основной логики.

Ключевым моментом работы программы является динамическая загрузка пользовательской разделяемой библиотеки `libs/libcalc1.so`. Системный вызов `newfstatat` подтверждает существование файла библиотеки, после чего он открывается с помощью `openat`, считывается его ELF-заголовок, и библиотека отображается в адресное пространство процесса через `mmap` с флагами `PROT\_READ|PROT\_EXEC`. Область памяти, содержащая данные библиотеки, дополнительно защищается с помощью `mprotect`, что обеспечивает корректное исполнение кода и защиту от несанкционированной записи в исполняемые сегменты. Это указывает на то, что программа использует механизм динамической загрузки (скорее всего, через `dlopen` и `dlsym` на уровне исходного кода), чтобы в рантайме подключать и вызывать функции из внешнего модуля.

После загрузки библиотеки программа выполняет операции ввода-вывода: с помощью `read(0, ...)` она считывает строку `"2 12 3\n"` из стандартного ввода, что свидетельствует о передаче аргументов или параметров вычисления. Затем, как следует из ответа, библиотека `libcalc1.so` обрабатывает эти данные и возвращает результат, который программа выводит в стандартный поток с помощью `write(1, "2.61304\n", 8)`. Этот вывод указывает на выполнение некоторой математической операции, возможно, вычисления значения функции или выражения, реализованного в пользовательской библиотеке. После этого программа пытается прочитать ещё один блок данных, но получает 0 байт, что означает конец ввода (EOF), и завершает свою работу.

На финальном этапе происходит выгрузка пользовательской библиотеки из памяти: вызов `munmap(0xffffa72e0000, 69680)` освобождает ранее отображённый сегмент, после чего процесс завершается с помощью `exit\_group(0)`, возвращая код успешного завершения. Отсутствие ошибок в системных вызовах, корректная последовательность загрузки и выгрузки библиотек, а также успешный ввод-вывод подтверждают, что приложение работает стабильно и в полной мере реализует задуманную архитектуру: динамическая загрузка внешних модулей в рантайме с последующей обработкой пользовательских данных и выдачей результата. Таким образом, лог `strace` демонстрирует корректное использование механизмов динамического связывания, управления памятью и ввода-вывода в рамках POSIX-совместимой системы.

**Вывод**

В ходе выполнения лабораторной работы №5 был проведён детальный анализ системных вызовов программ, разработанных в предыдущих четырёх лабораторных работах, с помощью утилиты strace. Это позволило наглядно увидеть, как высокоуровневые операции на языке Си (создание процессов, работа с каналами, разделяемой памятью, потоками и динамическими библиотеками) транслируются в низкоуровневые запросы к ядру операционной системы, такие как clone, pipe2, mmap, openat и munmap. Анализ подтвердил корректность реализации межпроцессного взаимодействия и управления ресурсами в каждой из рассмотренных программ, а также выявил характерные паттерны использования системных вызовов для разных механизмов коммуникации (pipes, shared memory) и многозадачности (процессы vs потоки).