Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №3 по курсу**

**«Операционные системы»**

Группа: М8О-209Б-22

Студент: Концебалов О.С.

Преподаватель: Пономарев Н.В.

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

Дата: 25.12.2023

Москва, 2023.

# Содержание

1. Постановка задачи.
2. Общие сведения о программе.
3. Общий метод и алгоритм решения.
4. Код программы.
5. Демонстрация работы программы.
6. Вывод.

## Постановка задачи

Составить и отладить программу на языке Си, осуществляющую работу с процессами и взаимодействие между ними в одной из двух операционных систем. В результате работы программа (основной процесс) должен создать для решение задачи один или несколько дочерних процессов. Взаимодействие между процессами осуществляется через системные сигналы/события и/или через отображаемые файлы (memory-mapped files). Необходимо обрабатывать системные ошибки, которые могут возникнуть в результате работы.

## Общие сведения о программе

Программа состоит из 3 папок: src, memory\_files, output\_files. В папке src Папка run\_files содержит два файла run.cpp и calculator.cpp с функцией main, которые будут скомпилированы в исполняемые. В output\_files находятся файлы полученные в результате работы программы. В memory\_files находится файл memory.txt, который используется в роли обменника данными между родительским и дочерним процессами. Также есть Makefile для удобной сборки и запуска программы.

## Общий метод и алгоритм решения

1. Программа main открывает файл для передачи данных и делает маппинг
2. Вводится семафор, чтобы заблокировать дочерний процесс
3. Происходит fork, после чего идет разделение родительского и дочернего процесса
4. Родительский процесс читает данные из входного потока и записывает их в memory map.
5. После окончания записи семафор разблокируется и родительский процесс ожидает завершения дочернего.
6. Дочерний процесс запускает calculator при помощи exec
7. Дочерний процесс открывает и маппирует общий файл
8. Как только семафор разблокирован, процесс читает данные из мапа и производит вычисления
9. Вывод записывается в выходной файл

## Код программы

./src/run.cpp

#include <fcntl.h>

#include <iostream>

#include <semaphore.h>

#include <stdio.h>

#include <sys/mman.h>

#include <sys/wait.h>

#include <unistd.h>

int main() {

    int memoryd;

    memoryd = open("memory\_files/memory.txt", O\_RDWR | O\_CREAT | O\_TRUNC, 0666);

    ftruncate(memoryd, 1024);

    char\* buffer = (char\*)mmap(NULL, 1024, PROT\_READ | PROT\_WRITE, MAP\_SHARED, memoryd, 0);

    close(memoryd);

    sem\_t\* \_sem = sem\_open("mmap\_sem11", O\_CREAT, S\_IRWXU, 0);

    sem\_t\* \_sem2 = sem\_open("mmap\_sem22", O\_CREAT, S\_IRWXU, 1);

    sem\_close(\_sem);

    sem\_close(\_sem2);

    sem\_unlink("mmap\_sem11");

    sem\_unlink("mmap\_sem22");

    sem\_t\* sem = sem\_open("mmap\_sem11", O\_CREAT, S\_IRWXU, 0);

    sem\_t\* sem2 = sem\_open("mmap\_sem22", O\_CREAT, S\_IRWXU, 1);

    if (sem == SEM\_FAILED) {

        perror("Could not open semaphore");

        return -1;

    }

    pid\_t id = fork();

    if (id == -1) {

        return 2;

    } else if (id == 0) {

        execl("build/src/\_calculator\_exe", "build/src/\_calculator\_exe", "mmap\_sem", NULL);

        return 3;

    } else {

        sem\_wait(sem2);

        char c;

        c = getchar();

        size\_t i = 0;

        while (c != EOF) {

            buffer[i++] = c;

            c = getchar();

        }

        buffer[i] = c;

        sem\_post(sem);

        munmap(buffer, 1024);

        int status;

        waitpid(0, &status, 0);

        if (status != 0)

            perror("Child process exited with an error");

        return status;

    }

}

./src/calculator.cpp

#include <fcntl.h>

#include <iostream>

#include <semaphore.h>

#include <stdio.h>

#include <sys/mman.h>

#include <unistd.h>

int main(int argc, char\*\* argv) {

    std::cout << "Child process started" << std::endl;

    FILE\* f = fopen("output\_files/output.txt", "w+");

    fprintf(f, "The sums are: ");

    int memoryd;

    memoryd = open("memory\_files/memory.txt", O\_RDWR, 0666);

    char\* buffer = (char\*)mmap(NULL, 1024, PROT\_READ | PROT\_WRITE, MAP\_SHARED, memoryd, 0);

    close(memoryd);

    sem\_t\* sem = sem\_open("mmap\_sem11", O\_CREAT, S\_IRWXU, 0);

    sem\_t\* sem2 = sem\_open("mmap\_sem22", O\_CREAT, S\_IRWXU, 1);

    if (sem == SEM\_FAILED) {

        perror("Could not open semaphore");

        return -1;

    }

    int num = 0, sum = 0;

    size\_t i = 0;

    sem\_wait(sem);

    while (buffer[i] != -1) {

        if (buffer[i] != ' ' && buffer[i] != '\n') {

            num \*= 10;

            num += buffer[i] - '0';

        } else if (buffer[i] == ' ') {

            sum += num;

            num = 0;

        } else if (buffer[i] == '\n') {

            if (num != 0)

                sum += num;

            fprintf(f, "%d ", sum);

            sum = 0;

            num = 0;

        }

        ++i;

    }

    sem\_post(sem2);

    sem\_close(sem);

    sem\_close(sem2);

    munmap(buffer, 1024);

    return 0;

}

## Использование утилиты strace

execve("./build/src/\_run\_exe", ["./build/src/\_run\_exe"], 0x7ffd963b0e48 /\* 60 vars \*/) = 0

brk(NULL) = 0x55b072e94000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fff7c79c290) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f98f3d52000

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=67103, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 67103, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f98f3d41000

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=2260296, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 2275520, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f98f3a00000

mprotect(0x7f98f3a9a000, 1576960, PROT\_NONE) = 0

mmap(0x7f98f3a9a000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7f98f3a9a000

mmap(0x7f98f3bab000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7f98f3bab000

mmap(0x7f98f3c1b000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7f98f3c1b000

mmap(0x7f98f3c29000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f98f3c29000

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\0 =\340\2563\265?\356\25x\261\27\313A#\350"..., 68, 896) = 68

newfstatat(3, "", {st\_mode=S\_IFREG|0755, 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) = 0x7f98f3600000

mmap(0x7f98f3628000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7f98f3628000

mmap(0x7f98f37bd000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7f98f37bd000

mmap(0x7f98f3815000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x214000) = 0x7f98f3815000

mmap(0x7f98f381b000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f98f381b000

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) = 0x7f98f3c5a000

mmap(0x7f98f3c68000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7f98f3c68000

mmap(0x7f98f3ce4000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7f98f3ce4000

mmap(0x7f98f3d3f000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7f98f3d3f000

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) = 0x7f98f3c3a000

mmap(0x7f98f3c3d000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f98f3c3d000

mmap(0x7f98f3c54000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7f98f3c54000

mmap(0x7f98f3c58000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7f98f3c58000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f98f3c38000

arch\_prctl(ARCH\_SET\_FS, 0x7f98f3c393c0) = 0

set\_tid\_address(0x7f98f3c39690) = 4863

set\_robust\_list(0x7f98f3c396a0, 24) = 0

rseq(0x7f98f3c39d60, 0x20, 0, 0x53053053) = 0

mprotect(0x7f98f3815000, 16384, PROT\_READ) = 0

mprotect(0x7f98f3c58000, 4096, PROT\_READ) = 0

mprotect(0x7f98f3d3f000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f98f3c36000

mprotect(0x7f98f3c1b000, 45056, PROT\_READ) = 0

mprotect(0x55b071223000, 4096, PROT\_READ) = 0

mprotect(0x7f98f3d8c000, 8192, PROT\_READ) = 0

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

munmap(0x7f98f3d41000, 67103) = 0

getrandom("\xfb\x88\xa3\x88\xd7\x7c\xcb\xd5", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x55b072e94000

brk(0x55b072eb5000) = 0x55b072eb5000

futex(0x7f98f3c2977c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

openat(AT\_FDCWD, "memory\_files/memory.txt", O\_RDWR|O\_CREAT|O\_TRUNC, 0666) = 3

ftruncate(3, 1024) = 0

mmap(NULL, 1024, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7f98f3d8b000

close(3) = 0

openat(AT\_FDCWD, "/dev/shm/sem.mmap\_sem11", O\_RDWR|O\_NOFOLLOW) = 3

newfstatat(3, "", {st\_mode=S\_IFREG|0700, st\_size=32, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7f98f3d51000

close(3) = 0

openat(AT\_FDCWD, "/dev/shm/sem.mmap\_sem22", O\_RDWR|O\_NOFOLLOW) = 3

newfstatat(3, "", {st\_mode=S\_IFREG|0700, st\_size=32, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7f98f3d50000

close(3) = 0

munmap(0x7f98f3d51000, 32) = 0

munmap(0x7f98f3d50000, 32) = 0

unlink("/dev/shm/sem.mmap\_sem11") = 0

unlink("/dev/shm/sem.mmap\_sem22") = 0

openat(AT\_FDCWD, "/dev/shm/sem.mmap\_sem11", O\_RDWR|O\_NOFOLLOW) = -1 ENOENT (No such file or directory)

getrandom("\xcb\x9e\x24\x73\x32\x15\x42\x52", 8, GRND\_NONBLOCK) = 8

newfstatat(AT\_FDCWD, "/dev/shm/sem.TdRjoX", 0x7fff7c79bf90, AT\_SYMLINK\_NOFOLLOW) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/dev/shm/sem.TdRjoX", O\_RDWR|O\_CREAT|O\_EXCL, 0700) = 3

write(3, "\0\0\0\0\0\0\0\0\200\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 32) = 32

mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7f98f3d51000

link("/dev/shm/sem.TdRjoX", "/dev/shm/sem.mmap\_sem11") = 0

newfstatat(3, "", {st\_mode=S\_IFREG|0700, st\_size=32, ...}, AT\_EMPTY\_PATH) = 0

unlink("/dev/shm/sem.TdRjoX") = 0

close(3) = 0

openat(AT\_FDCWD, "/dev/shm/sem.mmap\_sem22", O\_RDWR|O\_NOFOLLOW) = -1 ENOENT (No such file or directory)

getrandom("\xf2\x59\x8a\x6f\x91\x85\x1f\xd4", 8, GRND\_NONBLOCK) = 8

newfstatat(AT\_FDCWD, "/dev/shm/sem.u1QCCw", 0x7fff7c79bf90, AT\_SYMLINK\_NOFOLLOW) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/dev/shm/sem.u1QCCw", O\_RDWR|O\_CREAT|O\_EXCL, 0700) = 3

write(3, "\1\0\0\0\0\0\0\0\200\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 32) = 32

mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7f98f3d50000

link("/dev/shm/sem.u1QCCw", "/dev/shm/sem.mmap\_sem22") = 0

newfstatat(3, "", {st\_mode=S\_IFREG|0700, st\_size=32, ...}, AT\_EMPTY\_PATH) = 0

unlink("/dev/shm/sem.u1QCCw") = 0

close(3) = 0

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLDstrace: Process 4864 attached

, child\_tidptr=0x7f98f3c39690) = 4864

[pid 4864] set\_robust\_list(0x7f98f3c396a0, 24 <unfinished ...>

[pid 4863] newfstatat(0, "", <unfinished ...>

[pid 4864] <... set\_robust\_list resumed>) = 0

[pid 4863] <... newfstatat resumed>{st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0

[pid 4864] execve("build/src/\_calculator\_exe", ["build/src/\_calculator\_exe", "mmap\_sem"], 0x7fff7c79c468 /\* 60 vars \*/ <unfinished ...>

[pid 4863] read(0, <unfinished ...>

[pid 4864] <... execve resumed>) = 0

[pid 4864] brk(NULL) = 0x557b091a1000

[pid 4864] arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fff148dc8d0) = -1 EINVAL (Invalid argument)

[pid 4864] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdf3fa97000

[pid 4864] access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

[pid 4864] openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

[pid 4864] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=67103, ...}, AT\_EMPTY\_PATH) = 0

[pid 4864] mmap(NULL, 67103, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fdf3fa86000

[pid 4864] close(3) = 0

[pid 4864] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = 3

[pid 4864] 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

[pid 4864] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=2260296, ...}, AT\_EMPTY\_PATH) = 0

[pid 4864] mmap(NULL, 2275520, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdf3f800000

[pid 4864] mprotect(0x7fdf3f89a000, 1576960, PROT\_NONE) = 0

[pid 4864] mmap(0x7fdf3f89a000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7fdf3f89a000

[pid 4864] mmap(0x7fdf3f9ab000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7fdf3f9ab000

[pid 4864] mmap(0x7fdf3fa1b000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7fdf3fa1b000

[pid 4864] mmap(0x7fdf3fa29000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fdf3fa29000

[pid 4864] close(3) = 0

[pid 4864] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

[pid 4864] 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

[pid 4864] 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

[pid 4864] 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

[pid 4864] pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0 =\340\2563\265?\356\25x\261\27\313A#\350"..., 68, 896) = 68

[pid 4864] newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2216304, ...}, AT\_EMPTY\_PATH) = 0

[pid 4864] 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

[pid 4864] mmap(NULL, 2260560, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdf3f400000

[pid 4864] mmap(0x7fdf3f428000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7fdf3f428000

[pid 4864] mmap(0x7fdf3f5bd000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7fdf3f5bd000

[pid 4864] mmap(0x7fdf3f615000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x214000) = 0x7fdf3f615000

[pid 4864] mmap(0x7fdf3f61b000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fdf3f61b000

[pid 4864] close(3) = 0

[pid 4864] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libm.so.6", O\_RDONLY|O\_CLOEXEC) = 3

[pid 4864] 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

[pid 4864] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=940560, ...}, AT\_EMPTY\_PATH) = 0

[pid 4864] mmap(NULL, 942344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdf3f719000

[pid 4864] mmap(0x7fdf3f727000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7fdf3f727000

[pid 4864] mmap(0x7fdf3f7a3000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7fdf3f7a3000

[pid 4864] mmap(0x7fdf3f7fe000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7fdf3f7fe000

[pid 4864] close(3) = 0

[pid 4864] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3

[pid 4864] 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

[pid 4864] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=125488, ...}, AT\_EMPTY\_PATH) = 0

[pid 4864] mmap(NULL, 127720, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fdf3fa66000

[pid 4864] mmap(0x7fdf3fa69000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7fdf3fa69000

[pid 4864] mmap(0x7fdf3fa80000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7fdf3fa80000

[pid 4864] mmap(0x7fdf3fa84000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7fdf3fa84000

[pid 4864] close(3) = 0

[pid 4864] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdf3fa64000

[pid 4864] arch\_prctl(ARCH\_SET\_FS, 0x7fdf3fa653c0) = 0

[pid 4864] set\_tid\_address(0x7fdf3fa65690) = 4864

[pid 4864] set\_robust\_list(0x7fdf3fa656a0, 24) = 0

[pid 4864] rseq(0x7fdf3fa65d60, 0x20, 0, 0x53053053) = 0

[pid 4864] mprotect(0x7fdf3f615000, 16384, PROT\_READ) = 0

[pid 4864] mprotect(0x7fdf3fa84000, 4096, PROT\_READ) = 0

[pid 4864] mprotect(0x7fdf3f7fe000, 4096, PROT\_READ) = 0

[pid 4864] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fdf3fa62000

[pid 4864] mprotect(0x7fdf3fa1b000, 45056, PROT\_READ) = 0

[pid 4864] mprotect(0x557b07a53000, 4096, PROT\_READ) = 0

[pid 4864] mprotect(0x7fdf3fad1000, 8192, PROT\_READ) = 0

[pid 4864] prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

[pid 4864] munmap(0x7fdf3fa86000, 67103) = 0

[pid 4864] getrandom("\xc8\xa5\x13\x89\x66\x0f\x32\x35", 8, GRND\_NONBLOCK) = 8

[pid 4864] brk(NULL) = 0x557b091a1000

[pid 4864] brk(0x557b091c2000) = 0x557b091c2000

[pid 4864] futex(0x7fdf3fa2977c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

[pid 4864] newfstatat(1, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0

[pid 4864] write(1, "Child process started\n", 22Child process started

) = 22

[pid 4864] openat(AT\_FDCWD, "output\_files/output.txt", O\_RDWR|O\_CREAT|O\_TRUNC, 0666) = 3

[pid 4864] newfstatat(3, "", {st\_mode=S\_IFREG|0664, st\_size=0, ...}, AT\_EMPTY\_PATH) = 0

[pid 4864] openat(AT\_FDCWD, "memory\_files/memory.txt", O\_RDWR) = 4

[pid 4864] mmap(NULL, 1024, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 4, 0) = 0x7fdf3fad0000

[pid 4864] close(4) = 0

[pid 4864] openat(AT\_FDCWD, "/dev/shm/sem.mmap\_sem11", O\_RDWR|O\_NOFOLLOW) = 4

[pid 4864] newfstatat(4, "", {st\_mode=S\_IFREG|0700, st\_size=32, ...}, AT\_EMPTY\_PATH) = 0

[pid 4864] mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 4, 0) = 0x7fdf3fa96000

[pid 4864] close(4) = 0

[pid 4864] openat(AT\_FDCWD, "/dev/shm/sem.mmap\_sem22", O\_RDWR|O\_NOFOLLOW) = 4

[pid 4864] newfstatat(4, "", {st\_mode=S\_IFREG|0700, st\_size=32, ...}, AT\_EMPTY\_PATH) = 0

[pid 4864] mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 4, 0) = 0x7fdf3fa95000

[pid 4864] close(4) = 0

[pid 4864] futex(0x7fdf3fa96000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, FUTEX\_BITSET\_MATCH\_ANY1 2 3 4 5

<unfinished ...>

[pid 4863] <... read resumed>"1 2 3 4 5\n", 1024) = 10

[pid 4863] read(0, "", 1024) = 0

[pid 4863] futex(0x7f98f3d51000, FUTEX\_WAKE, 1) = 1

[pid 4864] <... futex resumed>) = 0

[pid 4863] munmap(0x7f98f3d8b000, 1024 <unfinished ...>

[pid 4864] munmap(0x7fdf3fa96000, 32) = 0

[pid 4863] <... munmap resumed>) = 0

[pid 4863] wait4(0, <unfinished ...>

[pid 4864] munmap(0x7fdf3fa95000, 32) = 0

[pid 4864] munmap(0x7fdf3fad0000, 1024) = 0

[pid 4864] write(3, "The sums are: 15 ", 17) = 17

[pid 4864] exit\_group(0) = ?

[pid 4864] +++ exited with 0 +++

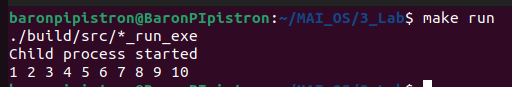
<... wait4 resumed>[{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 4864

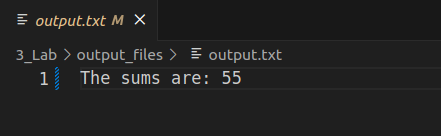
--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=4864, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

exit\_group(0) = ?

+++ exited with 0 +++

## Демонстрация работы программы





## Вывод

В процессе выполнения работы я познакомился с таким инструментом, как memory map. Memory map создает виртуальную копию файла, что позволяет при работе с фалами не использовать дорогие операции чтения из файла и записи в файл. При использовании memory map ответственность за соответствие файла и его мапа переносится на операционную систему. В процессе выполнения работы я столкнулся с определенными трудностями. Первоначально я не использовал семафор, поэтому пока происходил ввод данных, дочерний процесс уже пытался их прочитать, что приводило к ошибке. Также после добавления семафора, я добавил лишнее увеличение счетчика, из-за чего семафор работал только при первом запуске программы.