Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

Лабораторная работа №3 по курсу

«Операционные системы»

Группа: М8О-216БВ-24

Студент: Ходаков Павел

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 26.11.25

Москва, 2025

Постановка задачи

Вариант 7.

Родительский процесс создает дочерний процесс. Первой строчкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для обработки.

Родительский процесс: создает объект разделяемой памяти (shared memory) с использованием shm_open; устанавливает размер разделяемой памяти с помощью ftruncate; отображает разделяемую память в свое адресное пространство через mmap; создает два именованных семафора для синхронизации процессов; порождает дочерний процесс с помощью fork; записывает имя файла в разделяемую память; сигнализирует дочернему процессу через семафор о наличии задачи; читает результат вычислений из разделяемой памяти и выводит в стандартный вывод; после завершения работы освобождает ресурсы (munmap, shm_unlink, sem_unlink).

Ддочерний процесс открывает существующий объект разделяемой памяти через shm_open; отображает разделяемую память в свое адресное пространство через mmap; открывает существующие именованные семафоры; в цикле ожидает сигнала от родительского процесса через семафор; читает имя файла из разделяемой памяти; открывает указанный файл на чтение с помощью open; читает содержимое файла построчно; для каждой строки парсит числа с плавающей точкой, разделенные пробелами; вычисляет сумму чисел для каждой строки; записывает результаты в разделяемую память; сигнализирует родительскому процессу через семафор о готовности результата.

В файле записаны команды вида: «число число число<endline>». Дочерний процесс считает их сумму и выводит результат. Числа имеют тип float. Количество чисел может быть произвольным.

Общий метод и алгоритм решения

Использованные системные вызовы:

- `i`nt shm_open(const char *name, int oflag, mode_t mode) – создает/открывает объект разделяемой памяти
- int shm_unlink(const char *name) – удаляет объект разделяемой памяти
- int ftruncate(int fd, off_t length) – устанавливает размер разделяемой памяти
- void *mmap(void *addr, size_t length, int prot, int flags, int fd, off_t offset) – отображает разделяемую память в адресное пространство процесса
- int munmap(void *addr, size_t length) – удаляет отображение памяти
- sem_t *sem_open(const char *name, int oflag, mode_t mode, unsigned int value) – создает/открывает именованный семафор
- int sem_wait(sem_t *sem) – уменьшает значение семафора (ожидание)
- int sem_post(sem_t *sem) – увеличивает значение семафора (сигнал)
- int sem_close(sem_t *sem) – закрывает семафор

- int sem_unlink(const char *name) – удаляет семафор
- pid_t fork(void) – создает дочерний процесс
- int execl(const char *path, const char *arg, (char *) NULL) – заменяет код текущего процесса новой программой
- pid_t waitpid(pid_t pid, int *status, int options) – ожидает завершения процесса
- void exit(int status) – завершает процесс

В рамках лабораторной работы я изучил теоретический материал (материалы лекций и примеры кода, материалы из интернета по системным вызовам).

Родительский процесс (сервер):

1. Создает объект разделяемой памяти с помощью shm_open и устанавливает его размер через ftruncate
2. Отображает разделяемую память в свое адресное пространство с помощью mmap
3. Создает два именованных семафора для синхронизации с дочерним процессом
4. Порождает дочерний процесс с помощью fork
5. Посимвольно считывает из стандартного ввода имя файла до конца строки
6. Записывает имя файла в разделяемую память
7. Сигнализирует дочернему процессу через семафор о наличии новой задачи
8. Ожидает ответ от дочернего процесса через семафор
9. Читает результат вычислений из разделяемой памяти и выводит в стандартный вывод
10. После завершения работы освобождает ресурсы: удаляет отображение памяти, закрывает и удаляет семафоры и объект разделяемой памяти

Дочерний процесс (клиент):

1. Открывает существующий объект разделяемой памяти через shm_open
2. Отображает разделяемую память в свое адресное пространство через mmap
3. Открывает существующие именованные семафоры
4. В цикле ожидает сигнала от родительского процесса через семафор
5. Читает имя файла из разделяемой памяти
6. Открывает указанный файл на чтение с помощью open
7. Читает содержимое файла построчно с помощью read
8. Для каждой строки парсит числа с плавающей точкой, разделенные пробелами
9. Вычисляет сумму чисел для каждой строки

10.Записывает результаты в разделяемую память

11.Сигнализирует родительскому процессу через семафор о готовности результата

12.После завершения работы освобождает ресурсы и корректно завершается

Синхронизация процессов осуществляется через два именованных семафора, которые обеспечивают корректное взаимодействие между родительским и дочерним процессами без состояния гонки.

Код программы

https://github.com/n3tw4lk3r/MAI-OS-labs/tree/main/lab_3

Протокол работы программы

Вывод strace:

```
31712 11:09:18.741561 execve("./parent", ["./parent"], 0x7ffd58eb2980 /* 63 vars */) = 0
31712 11:09:18.742060 brk(NULL)         = 0x559b7e074000
31712 11:09:18.742235 access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
31712 11:09:18.742406 openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
31712 11:09:18.742555 fstat(3, {st_mode=S_IFREG|0644, st_size=153051, ...}) = 0
31712 11:09:18.742635 mmap(NULL, 153051, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fab95dcc000
31712 11:09:18.742705 close(3)          = 0
31712 11:09:18.742780 openat(AT_FDCWD, "/usr/lib/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = 3
31712 11:09:18.742847 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"..., 832) = 832
31712 11:09:18.742909 fstat(3, {st_mode=S_IFREG|0755, st_size=22480984, ...}) = 0
31712 11:09:18.742975 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
-1, 0) = 0x7fab95dca000
31712 11:09:18.743058 mmap(NULL, 2699336, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fab95a00000
31712 11:09:18.743132 mmap(0x7fab95a97000, 1396736, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x97000) = 0x7fab95a97000
31712 11:09:18.743203 mmap(0x7fab95bec000, 598016, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1ec000) = 0x7fab95bec000
31712 11:09:18.743266 mmap(0x7fab95c7e000, 73728, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x27e000) = 0x7fab95c7e000
31712 11:09:18.743341 mmap(0x7fab95c90000, 12360, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fab95c90000
31712 11:09:18.743419 close(3)          = 0
31712 11:09:18.743483 openat(AT_FDCWD, "/usr/lib/libm.so.6", O_RDONLY|O_CLOEXEC) = 3
31712 11:09:18.743553 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"..., 832) = 832
31712 11:09:18.743614 fstat(3, {st_mode=S_IFREG|0755, st_size=1100400, ...}) = 0
31712 11:09:18.743674 mmap(NULL, 1102152, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fab95cbc000
31712 11:09:18.743738 mmap(0x7fab95ccb000, 569344, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0xf000) = 0x7fab95ccb000
31712 11:09:18.743809 mmap(0x7fab95d56000, 466944, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x9a000) = 0x7fab95d56000
31712 11:09:18.743880 mmap(0x7fab95dc8000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x10b000) = 0x7fab95dc8000
31712 11:09:18.743960 close(3)          = 0
31712 11:09:18.744023 openat(AT_FDCWD, "/usr/lib/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3
31712 11:09:18.744100 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"..., 832) = 832
31712 11:09:18.744164 fstat(3, {st_mode=S_IFREG|0644, st_size=905800, ...}) = 0
```

```
31712 11:09:18.744228 mmap(NULL, 180712, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fab959d3000
31712 11:09:18.744289 mmap(0x7fab959d7000, 143360, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7fab959d7000
31712 11:09:18.744357 mmap(0x7fab959fa000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x27000) = 0x7fab959fa000
31712 11:09:18.744423 mmap(0x7fab959fe000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2a000) = 0x7fab959fe000
31712 11:09:18.744499 close(3)          = 0
31712 11:09:18.744557 openat(AT_FDCWD, "/usr/lib/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
31712 11:09:18.744620 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360w\2\0\0\0\0"..., 832) = 832
31712 11:09:18.744680 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"..., 896, 64) = 896
31712 11:09:18.744741 fstat(3, {st_mode=S_IFREG|0755, st_size=2145632, ...}) = 0
31712 11:09:18.744805 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"..., 896, 64) = 896
31712 11:09:18.744863 mmap(NULL, 2169904, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fab95600000
31712 11:09:18.744924 mmap(0x7fab95624000, 1511424, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x24000) = 0x7fab95624000
31712 11:09:18.744990 mmap(0x7fab95795000, 454656, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x195000) = 0x7fab95795000
31712 11:09:18.745077 mmap(0x7fab95804000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x203000) = 0x7fab95804000
31712 11:09:18.745154 mmap(0x7fab9580a000, 31792, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fab9580a000
31712 11:09:18.745226 close(3)          = 0
31712 11:09:18.745300 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
-1, 0) = 0x7fab95cba000
31712 11:09:18.745372 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
-1, 0) = 0x7fab95cb8000
31712 11:09:18.745432 arch_prctl(ARCH_SET_FS, 0x7fab95cbb680) = 0
31712 11:09:18.745488 set_tid_address(0x7fab95cbb950) = 31712
31712 11:09:18.745549 set_robust_list(0x7fab95cbb960, 24) = 0
31712 11:09:18.745605 rseq(0x7fab95cbb5a0, 0x20, 0, 0x53053053) = 0
31712 11:09:18.745733 mprotect(0x7fab95804000, 16384, PROT_READ) = 0
31712 11:09:18.745821 mprotect(0x7fab959fe000, 4096, PROT_READ) = 0
31712 11:09:18.745905 mprotect(0x7fab95dc8000, 4096, PROT_READ) = 0
31712 11:09:18.746687 mprotect(0x7fab95c7e000, 69632, PROT_READ) = 0
31712 11:09:18.746762 mprotect(0x559b67690000, 4096, PROT_READ) = 0
31712 11:09:18.746824 mprotect(0x7fab95e33000, 8192, PROT_READ) = 0
31712 11:09:18.746903 prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
31712 11:09:18.746983 getrandom("\x73\x4c\x41\x9a\x9c\xb6\xe5\x2d", 8, GRND_NONBLOCK) = 8
31712 11:09:18.747059 munmap(0x7fab95dcc000, 153051) = 0
31712 11:09:18.747198 futex(0x7fab95c906bc, FUTEX_WAKE_PRIVATE, 2147483647) = 0
31712 11:09:18.747301 brk(NULL)         = 0x559b7e074000
31712 11:09:18.747356 brk(0x559b7e0a7000) = 0x559b7e0a7000
31712 11:09:18.747438 openat(AT_FDCWD, "/dev/shm/calc_shm", O_RDWR|O_CREAT|O_NOFOLLOW|
O_CLOEXEC, 0666) = 3
31712 11:09:18.747536 ftruncate(3, 514) = 0
31712 11:09:18.747599 mmap(NULL, 514, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7fab95df1000
31712 11:09:18.747674 openat(AT_FDCWD, "/dev/shm/sem.calc_sem_server", O_RDWR|O_NOFOLLOW|
O_CLOEXEC) = -1 ENOENT (No such file or directory)
31712 11:09:18.747749 rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
31712 11:09:18.747821 mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_DROPPABLE|
MAP_ANONYMOUS, -1, 0) = 0x7fab95df0000
31712 11:09:18.747882 mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
-1, 0) = 0x7fab95def000
31712 11:09:18.747948 rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
```

```
31712 11:09:18.748024 getrandom("\x99\x27\x86\xc6\x8c\xf7\xa4\x39\xd4\xdf\xa4\x61\xc4\xab\x7e\x27\xeb\xcd\x62\xab\x70\xae\xff\xa0\x33\x49\x86\x57\xc3\xb6\xcc\x9f", 32, 0) = 32
31712 11:09:18.748116 newfstatat(AT_FDCWD, "/dev/shm/sem.Ct25z7", 0x7fff018d7d70, AT_SYMLINK_NOFOLLOW) = -1 ENOENT (No such file or directory)
31712 11:09:18.748222 openat(AT_FDCWD, "/dev/shm/sem.Ct25z7", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC, 0666) = 4
31712 11:09:18.748313 write(4, "\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
31712 11:09:18.748395 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7fab95dee000
31712 11:09:18.748472 link("/dev/shm/sem.Ct25z7", "/dev/shm/sem.calc_sem_server") = 0
31712 11:09:18.748560 fstat(4, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0
31712 11:09:18.748640 unlink("/dev/shm/sem.Ct25z7") = 0
31712 11:09:18.748735 close(4)          = 0
31712 11:09:18.748807 openat(AT_FDCWD, "/dev/shm/sem.calc_sem_client", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = -1 ENOENT (No such file or directory)
31712 11:09:18.748914 newfstatat(AT_FDCWD, "/dev/shm/sem.7tE35e", 0x7fff018d7d70, AT_SYMLINK_NOFOLLOW) = -1 ENOENT (No such file or directory)
31712 11:09:18.749014 openat(AT_FDCWD, "/dev/shm/sem.7tE35e", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC, 0666) = 4
31712 11:09:18.749125 write(4, "\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
31712 11:09:18.749219 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7fab95ded000
31712 11:09:18.749342 link("/dev/shm/sem.7tE35e", "/dev/shm/sem.calc_sem_client") = 0
31712 11:09:18.749445 fstat(4, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0
31712 11:09:18.749535 unlink("/dev/shm/sem.7tE35e") = 0
31712 11:09:18.749618 close(4)          = 0
31712 11:09:18.749681 rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
31712 11:09:18.749745 clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0x7fab95cbb950) = 31713
31712 11:09:18.749952 rt_sigprocmask(SIG_SETMASK, [] <unfinished ...>
31713 11:09:18.749971 set_robust_list(0x7fab95cbb960, 24 <unfinished ...>
31712 11:09:18.749997 <... rt_sigprocmask resumed>, NULL, 8) = 0
31713 11:09:18.750025 <... set_robust_list resumed>) = 0
31712 11:09:18.750057 write(1, "Enter filename: ", 16) = 16
31713 11:09:18.750119 rt_sigprocmask(SIG_SETMASK, [] <unfinished ...>
31712 11:09:18.750143 read(0 <unfinished ...>
31713 11:09:18.750156 <... rt_sigprocmask resumed>, NULL, 8) = 0
31713 11:09:18.750227 munmap(0x7fab95df1000, 514) = 0
31713 11:09:18.750291 close(3)          = 0
31713 11:09:18.750366 execve("./child", ["./child"], 0x7fff018d8298 /* 63 vars */) = 0
31713 11:09:18.750759 brk(NULL)         = 0x55696f552000
31713 11:09:18.750846 access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
31713 11:09:18.750926 openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
31713 11:09:18.750998 fstat(3, {st_mode=S_IFREG|0644, st_size=153051, ...}) = 0
31713 11:09:18.751095 mmap(NULL, 153051, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fbcd5fef000
31713 11:09:18.751166 close(3)          = 0
31713 11:09:18.751245 openat(AT_FDCWD, "/usr/lib/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = 3
31713 11:09:18.751309 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"..., 832) = 832
31713 11:09:18.751369 fstat(3, {st_mode=S_IFREG|0755, st_size=22480984, ...}) = 0
31713 11:09:18.751429 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fbcd5fed000
31713 11:09:18.751492 mmap(NULL, 2699336, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fbcd5c00000
31713 11:09:18.751554 mmap(0x7fbcd5c97000, 1396736, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x97000) = 0x7fbcd5c97000
31713 11:09:18.751618 mmap(0x7fbcd5dec000, 598016, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1ec000) = 0x7fbcd5dec000
```

31713 11:09:18.751680 mmap(0x7fbcd5e7e000, 73728, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x27e000) = 0x7fbcd5e7e000

31713 11:09:18.751753 mmap(0x7fbcd5e90000, 12360, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fbcd5e90000

31713 11:09:18.751826 close(3) = 0

31713 11:09:18.751887 openat(AT_FDCWD, "/usr/lib/libm.so.6", O_RDONLY|O_CLOEXEC) = 3

31713 11:09:18.751953 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"..., 832) = 832

31713 11:09:18.752014 fstat(3, {st_mode=S_IFREG|0755, st_size=1100400, ...}) = 0

31713 11:09:18.752085 mmap(NULL, 1102152, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fbcd5edf000

31713 11:09:18.752151 mmap(0x7fbcd5eee000, 569344, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xf000) = 0x7fbcd5eee000

31713 11:09:18.752218 mmap(0x7fbcd5f79000, 466944, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9a000) = 0x7fbcd5f79000

31713 11:09:18.752281 mmap(0x7fbcd5feb000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x10b000) = 0x7fbcd5feb000

31713 11:09:18.752358 close(3) = 0

31713 11:09:18.752414 openat(AT_FDCWD, "/usr/lib/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3

31713 11:09:18.752477 read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

31713 11:09:18.752541 fstat(3, {st_mode=S_IFREG|0644, st_size=905800, ...}) = 0

31713 11:09:18.752602 mmap(NULL, 180712, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fbcd5eb2000

31713 11:09:18.752667 mmap(0x7fbcd5eb6000, 143360, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7fbcd5eb6000

31713 11:09:18.752734 mmap(0x7fbcd5ed9000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x27000) = 0x7fbcd5ed9000

31713 11:09:18.752795 mmap(0x7fbcd5edd000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2a000) = 0x7fbcd5edd000

31713 11:09:18.752875 close(3) = 0

31713 11:09:18.752933 openat(AT_FDCWD, "/usr/lib/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

31713 11:09:18.752996 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360w\2\0\0\0\0\0"..., 832) = 832

31713 11:09:18.753074 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"..., 896, 64) = 896

31713 11:09:18.753139 fstat(3, {st_mode=S_IFREG|0755, st_size=2145632, ...}) = 0

31713 11:09:18.753206 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"..., 896, 64) = 896

31713 11:09:18.753265 mmap(NULL, 2169904, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fbcd5800000

31713 11:09:18.753327 mmap(0x7fbcd5824000, 1511424, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x24000) = 0x7fbcd5824000

31713 11:09:18.753390 mmap(0x7fbcd5995000, 454656, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x195000) = 0x7fbcd5995000

31713 11:09:18.753453 mmap(0x7fbcd5a04000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x203000) = 0x7fbcd5a04000

31713 11:09:18.753522 mmap(0x7fbcd5a0a000, 31792, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fbcd5a0a000

31713 11:09:18.753594 close(3) = 0

31713 11:09:18.753666 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fbcd5eb0000

31713 11:09:18.753736 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fbcd5eae000

31713 11:09:18.753798 arch_prctl(ARCH_SET_FS, 0x7fbcd5eb1680) = 0

31713 11:09:18.753852 set_tid_address(0x7fbcd5eb1950) = 31713

31713 11:09:18.753906 set_robust_list(0x7fbcd5eb1960, 24) = 0

31713 11:09:18.753961 rseq(0x7fbcd5eb15a0, 0x20, 0, 0x53053053) = 0

31713 11:09:18.754149 mprotect(0x7fbcd5a04000, 16384, PROT_READ) = 0

31713 11:09:18.754254 mprotect(0x7fbcd5edd000, 4096, PROT_READ) = 0

31713 11:09:18.754345 mprotect(0x7fbcd5feb000, 4096, PROT_READ) = 0

31713 11:09:18.755010 mprotect(0x7fbcd5e7e000, 69632, PROT_READ) = 0

31713 11:09:18.755119 mprotect(0x556933869000, 4096, PROT_READ) = 0

31713 11:09:18.755189 mprotect(0x7fbcd6056000, 8192, PROT_READ) = 0

31713 11:09:18.755266 prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
31713 11:09:18.755347 getrandom("\x2e\x55\xa2\xcd\xca\x55\x4a\x59", 8, GRND_NONBLOCK) = 8
31713 11:09:18.755412 munmap(0x7fbcd5fef000, 153051) = 0
31713 11:09:18.755538 futex(0x7fbcd5e906bc, FUTEX_WAKE_PRIVATE, 2147483647) = 0
31713 11:09:18.755633 brk(NULL) = 0x55696f552000
31713 11:09:18.755689 brk(0x55696f585000) = 0x55696f585000
31713 11:09:18.755770 openat(AT_FDCWD, "/dev/shm/calc_shm", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = 3
31713 11:09:18.755837 mmap(NULL, 514, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7fbcd6014000
31713 11:09:18.755962 fstat(4, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0
31713 11:09:18.756022 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7fbcd6013000
31713 11:09:18.756101 close(4) = 0
31713 11:09:18.756158 openat(AT_FDCWD, "/dev/shm/sem.calc_sem_client", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = 4
31713 11:09:18.756222 fstat(4, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0
31713 11:09:18.756282 mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7fbcd6012000
31713 11:09:18.756341 close(4) = 0
31713 11:09:18.756399 futex(0x7fbcd6012000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
31712 11:09:20.684575 <... read resumed>, "t", 1) = 1
31712 11:09:20.684870 read(0, "e", 1) = 1
31712 11:09:20.685211 read(0, "s", 1) = 1
31712 11:09:20.685492 read(0, "t", 1) = 1
31712 11:09:20.685691 read(0, ".", 1) = 1
31712 11:09:20.685876 read(0, "t", 1) = 1
31712 11:09:20.686073 read(0, "x", 1) = 1
31712 11:09:20.686177 read(0, "t", 1) = 1
31712 11:09:20.686273 read(0, "\n", 1) = 1

31713 11:09:20.686537 <... futex resumed>) = 0
31712 11:09:20.686587 futex(0x7fab95dee000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
31713 11:09:20.686611 openat(AT_FDCWD, "test.txt", O_RDONLY) = 4
31713 11:09:20.686721 read(4, "1.25 2.3 3.7\r\n2.65\r\n5.55 6.77 7\r"..., 8192) = 82
31713 11:09:20.686850 read(4, "", 8192) = 0
31713 11:09:20.686942 close(4) = 0
31713 11:09:20.687080 futex(0x7fbcd6013000, FUTEX_WAKE, 1 <unfinished ...>
31712 11:09:20.687127 <... futex resumed>) = 0
31713 11:09:20.687143 <... futex resumed>) = 1
31712 11:09:20.687159 write(1, "Result: ", 8 <unfinished ...>
31713 11:09:20.687184 futex(0x7fbcd6012000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
31712 11:09:20.687201 <... write resumed>) = 8
31712 11:09:20.687240 write(1, "7.25 2.65 19.32 1500.00 600.80 1"..., 38) = 38
31712 11:09:20.687324 write(1, "\n", 1) = 1
31712 11:09:20.687402 futex(0x7fab95ded000, FUTEX_WAKE, 1) = 1
31713 11:09:20.687464 <... futex resumed>) = 0
31712 11:09:20.687481 wait4(31713 <unfinished ...>
31713 11:09:20.687515 munmap(0x7fbcd6013000, 32) = 0
31713 11:09:20.687626 munmap(0x7fbcd6012000, 32) = 0
31713 11:09:20.687707 munmap(0x7fbcd6014000, 514) = 0

```
31713 11:09:20.687779 close(3)          = 0
31713 11:09:20.687891 exit_group(0)     = ?
31713 11:09:20.688305 +++ exited with 0 +++
31712 11:09:20.688328 <... wait4 resumed>, NULL, 0, NULL) = 31713
31712 11:09:20.688369 --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=31713, si_uid=1000,
si_status=0, si_utime=0, si_stime=0} ---
31712 11:09:20.688414 munmap(0x7fab95df1000, 514) = 0
31712 11:09:20.688493 close(3)          = 0
31712 11:09:20.688566 unlink("/dev/shm/calc_shm") = 0
31712 11:09:20.688684 munmap(0x7fab95dee000, 32) = 0
31712 11:09:20.688758 munmap(0x7fab95ded000, 32) = 0
31712 11:09:20.688833 unlink("/dev/shm/sem.calc_sem_server") = 0
31712 11:09:20.688915 unlink("/dev/shm/sem.calc_sem_client") = 0
31712 11:09:20.689082 exit_group(0)     = ?
31712 11:09:20.689387 +++ exited with 0 +++
```

Вывод

У меня успешно получилось реализовать программу на языке C++, которая использует межпроцессорное взаимодействие через shared memory и memory mapping. Все системные вызовы работают корректно,

программа завершается без ошибок. Трудностей в работе не возникло.