**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: М80-203Б-22

Студент: Касумова Н.Р.

Преподаватель: Миронов Е.С.

Оценка:

Подпись:

Дата: 27.12.23

**Цель рыботы:**

Приобретение практических навыков в:

* Освоение принципов работы с файловыми системами
* Обеспечение обмена данных между процессами посредством технологии «File mapping»

**Постановка задачи**

Вариант 7

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

Изображение выглядит как текст, диаграмма, круг, снимок экрана

Автоматически созданное описание

Системные вызовы:

* int shm\_open(const char \*name, int flag, mode\_t mode) – открыть общий объект памяти. Функция устанавливает соединение между объектом общей памяти и файловым дескриптором.

Int flag – режим доступа, mode\_t mode

* void mmap(void \*start, size\_t length, int prot, int flags, int fd, off\_t offset) – отражает файлы или устройства в памяти. Первый аргумент указывает на адрес в памяти, второй – length байтов, то есть длина в байтах, третий – желаемый режим защиты памяти, четвертый – тип отражаемого объекта, а также указывает принадлежат ли отраженные данные только этому процессу или их могут читать другие, пятый – описатель файла, а шестой – смещение файла
* int mumap(void \*start, size\_t length) – снимает отражение памяти файла. Первый аргумент указывает на адрес в памяти, а второй на длину в байтах
* shm\_unlink(const char \*name) – удаление имени объекта общей памяти
* ftruncate - укорачивает файл до указанной длины

**Код программы**

main.c

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <sys/mman.h>

#include <string.h>

#include <fcntl.h>

#define MAX\_SIZE 100000

int main() {

char filename[MAX\_SIZE];

printf("Введите имя файла для чтения: ");

fgets(filename, sizeof(filename), stdin);

int len = strlen(filename);

if (filename[len - 1] == '\n') {

filename[len - 1] = '\0';

}

int fd = shm\_open("/my\_shared\_memory", O\_CREAT | O\_RDWR | S\_IRUSR | S\_IWUSR);

if (fd == -1) {

perror("Ошибка при создании shared memory");

exit(EXIT\_FAILURE);

}

if(ftruncate(fd, sizeof(float)) == -1) {

perror("Ошибка");

exit(EXIT\_FAILURE);

}

float \*shared\_data = (float\* )mmap(NULL, sizeof(float), PROT\_READ | PROT\_WRITE, MAP\_SHARED, fd, 0);

pid\_t child\_id = fork();

if (child\_id == -1) {

perror("Ошибка при создании дочернего процесса");

exit(-1);

}

if (child\_id == 0) { //child

close(fd);

exit(EXIT\_SUCCESS);

if(execlp("./child", "./child", filename, NULL) == -1) {

perror("Ошибка при запуске дочернего процесса");

exit(-1);

};

}

else {

FILE \*file = fopen(filename, "r");

if (file == NULL) {

perror("Ошибка открытия файла");

exit(-1);

}

float number;

while(fscanf(file, "%f", &number) != EOF) {

\*shared\_data += number;

}

fclose(file);

wait(NULL);

printf("Сумма чисел из файла %s: %f\n", filename, \*shared\_data);

munmap(shared\_data, sizeof(float));

shm\_unlink("/my\_shared\_memory");

}

return 0;

}

Child.c

#include <stdio.h>

#include <stdlib.h>

#include <sys/mman.h>

#include <unistd.h>

#include <sys/types.h>

#include <fcntl.h>

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

if (argc != 2) {

fprintf(stderr, "Использование: %s <имя файла>\n", argv[0]);

exit(-1);

}

char \*filename = argv[1];

FILE \*file = fopen(filename, "r");

if (file == NULL) {

perror("Ошибка при открытии файла");

exit(-1);

}

float sum = 0, number;

int fd = shm\_open("/my\_shared\_memory", O\_RDWR, S\_IRWXU);

float \*shared\_data = (float\* )mmap(NULL, sizeof(float), PROT\_READ | PROT\_WRITE, MAP\_SHARED, fd, 0);

close(fd);

while (fscanf(file, "%f", &number) != EOF) {

sum += number;

}

fclose(file);

\*shared\_data = sum;

munmap(shared\_data, sizeof(float));

return 0;

}

**Strace**

ksu@DESKTOP-4J8FFR6:~/naida$ strace -f ./parent1

execve("./parent1", ["./parent1"], 0x7ffd78264648 /\* 22 vars \*/) = 0

brk(NULL) = 0x555a5c1b5000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffc5be03290) = -1 EINVAL (Invalid argument)

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=57411, ...}) = 0

mmap(NULL, 57411, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f5398e77000

close(3) = 0

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

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300A\2\0\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\346x\376T\245\322\302\t/\216G\353\v3\20^"..., 68, 880) = 68

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2029592, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f5398e75000

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\346x\376T\245\322\302\t/\216G\353\v3\20^"..., 68, 880) = 68

mmap(NULL, 2037344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f5398c83000

mmap(0x7f5398ca5000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x22000) = 0x7f5398ca5000

mmap(0x7f5398e1d000, 319488, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19a000) = 0x7f5398e1d000

mmap(0x7f5398e6b000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7f5398e6b000

mmap(0x7f5398e71000, 13920, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f5398e71000

close(3) = 0

arch\_prctl(ARCH\_SET\_FS, 0x7f5398e76540) = 0

mprotect(0x7f5398e6b000, 16384, PROT\_READ) = 0

mprotect(0x555a5a88c000, 4096, PROT\_READ) = 0

mprotect(0x7f5398eb3000, 4096, PROT\_READ) = 0

munmap(0x7f5398e77000, 57411) = 0

fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}) = 0

brk(NULL) = 0x555a5c1b5000

brk(0x555a5c1d6000) = 0x555a5c1d6000

fstat(0, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}) = 0

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\270\320\274\321\217 \321\204\320\260\320\271\320\273\320\260"..., 54Введите имя файла для чтения: ) = 54

read(0, task.txt

"\320\265\321\204task.txt\n", 1024) = 13

pipe([3, 4]) = 0

clone(strace: Process 78 attached

child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f5398e76810) = 78

[pid 78] close(3 <unfinished ...>

[pid 76] close(4 <unfinished ...>

[pid 78] <... close resumed> ) = 0

[pid 76] <... close resumed> ) = 0

[pid 78] dup2(4, 1 <unfinished ...>

[pid 76] read(3, <unfinished ...>

[pid 78] <... dup2 resumed> ) = 1

[pid 78] close(4) = 0

[pid 78] execve("./child1", ["./child1", "\320\265\321\204task.txt"], 0x7ffc5be03378 /\* 22 vars \*/) = 0

[pid 78] brk(NULL) = 0x55d593845000

[pid 78] arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fff84cee180) = -1 EINVAL (Invalid argument)

[pid 78] access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

[pid 78] openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

[pid 78] fstat(3, {st\_mode=S\_IFREG|0644, st\_size=57411, ...}) = 0

[pid 78] mmap(NULL, 57411, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fe2f0238000

[pid 78] close(3) = 0

[pid 78] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

[pid 78] read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300A\2\0\0\0\0\0"..., 832) = 832

[pid 78] 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 78] pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

[pid 78] pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\346x\376T\245\322\302\t/\216G\353\v3\20^"..., 68, 880) = 68

[pid 78] fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2029592, ...}) = 0

[pid 78] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe2f0236000

[pid 78] 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 78] pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

[pid 78] pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\346x\376T\245\322\302\t/\216G\353\v3\20^"..., 68, 880) = 68

[pid 78] mmap(NULL, 2037344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe2f0044000

[pid 78] mmap(0x7fe2f0066000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x22000) = 0x7fe2f0066000

[pid 78] mmap(0x7fe2f01de000, 319488, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19a000) = 0x7fe2f01de000

[pid 78] mmap(0x7fe2f022c000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7fe2f022c000

[pid 78] mmap(0x7fe2f0232000, 13920, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fe2f0232000

[pid 78] close(3) = 0

[pid 78] arch\_prctl(ARCH\_SET\_FS, 0x7fe2f0237540) = 0

[pid 78] mprotect(0x7fe2f022c000, 16384, PROT\_READ) = 0

[pid 78] mprotect(0x55d59207b000, 4096, PROT\_READ) = 0

[pid 78] mprotect(0x7fe2f0274000, 4096, PROT\_READ) = 0

[pid 78] munmap(0x7fe2f0238000, 57411) = 0

[pid 78] brk(NULL) = 0x55d593845000

[pid 78] brk(0x55d593866000) = 0x55d593866000

[pid 78] openat(AT\_FDCWD, "\320\265\321\204task.txt", O\_RDONLY) = -1 ENOENT (No such file or directory)

[pid 78] dup(2) = 3

[pid 78] fcntl(3, F\_GETFL) = 0x8002 (flags O\_RDWR|O\_LARGEFILE)

[pid 78] fstat(3, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}) = 0

[pid 78] write(3, "\320\236\321\210\320\270\320\261\320\272\320\260 \320\277\321\200\320\270 \320\276\321\202\320\272\321\200\321\213\321\202"..., 75Ошибка при открытии файла: No such file or directory

) = 75

[pid 78] close(3) = 0

[pid 78] exit\_group(-1) = ?

[pid 76] <... read resumed> "", 100000) = 0

[pid 78] +++ exited with 255 +++

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=78, si\_uid=1000, si\_status=255, si\_utime=0, si\_stime=1} ---

close(3) = 0

wait4(-1, NULL, 0, NULL) = 78

exit\_group(0) = ?

+++ exited with 0 +++

ksu@DESKTOP-4J8FFR6:~/naida$ strace ./parent1

execve("./parent1", ["./parent1"], 0x7ffd4ebdd820 /\* 22 vars \*/) = 0

brk(NULL) = 0x55575e858000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffd7cc65170) = -1 EINVAL (Invalid argument)

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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=57411, ...}) = 0

mmap(NULL, 57411, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fbdc6d60000

close(3) = 0

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

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300A\2\0\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\346x\376T\245\322\302\t/\216G\353\v3\20^"..., 68, 880) = 68

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2029592, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fbdc6d5e000

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\346x\376T\245\322\302\t/\216G\353\v3\20^"..., 68, 880) = 68

mmap(NULL, 2037344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fbdc6b6c000

mmap(0x7fbdc6b8e000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x22000) = 0x7fbdc6b8e000

mmap(0x7fbdc6d06000, 319488, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19a000) = 0x7fbdc6d06000

mmap(0x7fbdc6d54000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7fbdc6d54000

mmap(0x7fbdc6d5a000, 13920, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fbdc6d5a000

close(3) = 0

arch\_prctl(ARCH\_SET\_FS, 0x7fbdc6d5f540) = 0

mprotect(0x7fbdc6d54000, 16384, PROT\_READ) = 0

mprotect(0x55575d475000, 4096, PROT\_READ) = 0

mprotect(0x7fbdc6d9c000, 4096, PROT\_READ) = 0

munmap(0x7fbdc6d60000, 57411) = 0

fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}) = 0

brk(NULL) = 0x55575e858000

brk(0x55575e879000) = 0x55575e879000

fstat(0, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}) = 0

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\270\320\274\321\217 \321\204\320\260\320\271\320\273\320\260"..., 54Введите имя файла для чтения: ) = 54

read(0, task.txt

"task.txt\n", 1024) = 9

pipe([3, 4]) = 0

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7fbdc6d5f810) = 98

close(4) = 0

read(3, "\320\241\321\203\320\274\320\274\320\260 \321\207\320\270\321\201\320\265\320\273 \320\270\320\267 \321\204\320\260\320"..., 100000) = 58

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=98, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

write(1, "\320\241\321\203\320\274\320\274\320\260 \321\207\320\270\321\201\320\265\320\273 \320\270\320\267 \321\204\320\260\320"..., 58Сумма чисел из файла task.txt: 38.410999

) = 58

read(3, "", 100000) = 0

close(3) = 0

wait4(-1, NULL, 0, NULL) = 98

exit\_group(0) = ?

+++ exited with 0 +++

**Пример работы**

n@MacBook-Pro-N lab\_3 % ./parent

Введите имя файла для чтения: task1.txt

Сумма чисел из файла task1.txt: 37.426003

**Вывод**

В ходе выполнения лабораторной работы я научилась работать с системным вызовом mmap и применять в качестве средства взаимодействия между процессами. Для этого предоставили доступ к общей памяти. Выполнила первую лабораторную, используя вместо pipe’ов mmap, тем самым рассмотрела различные способы взаимодействия нескольких процессов. Мне показалось интересным выполнить еще раз выполнить первую лабораторную работу, но уже другим способом.