Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №4 по курсу

«Операционные системы»

Группа: М80-207Б-20

Студент: Мерц С.П.

Вариант: 12

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 25.12.21

Москва, 2021.

Содержание

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.

Вариант:

Программа состоит из файлов parent.c, child1.c, child2.c. В файле parent.c храниться родительский процесс и создание дочернего процесса, также перенаправление потока вывода. В child1.c. мы задаем первому ребенку перевести строки в верхний регистр. В child2.c. мы задаём убрать все двойные пробелы.

Программа использует следующие системные вызовы:

1. **read –** для чтения данных из входного потока.
2. **write** – для записи данных в файл или выходной поток.
3. **pipe –** для создания канала, через который процессы могут обмениваться информацией.
4. **fork –** для создания дочернего процесса.
5. **dup2** – для перенаправления потока вывода.
6. **execve** - замена образа памяти процесса

Общие сведения о программе

Программа состоит из 4х файлов: parent.c, child1.c, child2.c и mltshr.h. Ключевым файлом является mltshr.h так, как именно в нем происходят все основные задачи программы.

Программа использует следующие системные вызовы:

shm_open – для создания нового именнованного семафора.

shm_unlink – для удаления именованного семафора.

shm_destroy – для уничтожения семафора.

open - для создания файла и его открытия.

close – для закрытия файлового дескриптора.

mmap – для отображения файла в память.

fork – для создания дочернего процесса.

cond_wait – для блокировки семафора.

mutex_unlock – для разблокировки семафора.

dup2 – для перенаправления потока вывода.

Общий метод и алгоритм решения

Наша задача в данной лабораторной работе заключается в том, что у нас имеется не синхронизированный доступ к общему ресурсу. Мы бы хотели сделать так, чтобы на время работы с ресурсом к ним имел доступ только один поток, а остальные ждали, пока ресурс освободится. Это так называемое mutual exclusion – взаимное исключение, случай, когда необходимо удостовериться в том, что два (и более…) конкурирующих потока не находятся в критической секции кода одновременно.

В библиотеке pthreads один из методов разрешить эту ситуацию – это мьютексы. Мьютекс – это объект, который может находиться в двух состояниях. Он либо заблокирован (занят, залочен, захвачен) каким-то потоком, либо свободен. Поток, который захватил мьютекс, работает с участком кода. Остальные потоки, когда достигают мьютекса, ждут его разблокировки. Разблокировать мьютекс может только тот поток, который его захватил. Обычно освобождение занятого мьютекса происходит после исполнения критичного к совместному доступу участка кода.

Мьютекс – это экземпляр типа pthread_mutex_t. Перед использованием необходимо инициализировать мьютекс функцией pthread_mutex_init

int pthread_mutex_init(pthread_mutex_t *mutex, const pthread_mutexattr_t *attr);

где первый аргумент – указатель на мьютекс, а второй – аттрибуты мьютекса. Если указан NULL, то используются атрибуты по умолчанию. В случае удачной инициализации мьютекс переходит в состояние «инициализированный и свободный», а функция возвращает 0. Повторная инициализация инициализированного мьютекса приводит к неопределённому поведению.

Если мьютекс создан статически и не имеет дополнительных параметров, то он может быть инициализирован с помощью макроса PTHREAD_MUTEX_INITIALIZER

После использования мьютекса его необходимо уничтожить с помощью функции

int pthread_mutex_destroy(pthread_mutex_t *mutex);

В результате функция возвращает 0 в случае успеха или может возвратить код ошибки.

После создания мьютекса он может быть захвачен с помощью функции

int pthread_mutex_lock(pthread_mutex_t *mutex);

После этого участок кода становится недоступным остальным потокам – их выполнение блокируется до тех пор, пока мьютекс не будет освобождён. Освобождение должен провести поток, заблокировавший мьютекс, вызовом

int pthread_mutex_unlock(pthread_mutex_t *mutex);

При использовании мьютекса исполнение защищённого участка кода происходит последовательно всеми потоками, а не параллельно. Порядок доступа отдельных потоков не определён. Напишем теперь реализацию, в которой мьютекс будет передаваться в качестве параметра функции.

Хочется обратить внимание, что мьютекс один на всех. Если бы у каждого потока был свой собственный мьютекс, то они бы не блокировали работу друг друга.

Код программы

mltshr.h:

```
#ifndef MLTSHR

#define MLTSHR

#define MLTSHR_BLOCK_SIZE 4096


#include <stdio.h>

#include <sys/mman.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <sys/stat.h>

#include <pthread.h>

#include <stdlib.h>


typedef struct mltshr

{

	char memname[20]; // максимальная длина имени пайпа
```

```
	int memFd;  // создание переменной файлого дисриптора
	char* buffer;
} mltshr;


typedef struct state
{
	pthread_mutex_t mutex; // защищает отдельно, чтобы несколько потокоы не читали одновременно
	pthread_mutex_t write_mutex; // защащает отдельно, чтобы несколько потоков не писали одновременно
	pthread_cond_t cond; //условная переменная,для синхранихации между потоками
	int msglen; // кол-во байт, которое нужно отобразить в память
} state;


int mltshr_create(mltshr *ms, char *memname, char host)
{
	ms->memFd = shm_open(memname, O_CREAT | O_RDWR, S_IWUSR | S_IRUSR); // определяем обственно создаваемый объект разделяемой памяти для создания или открытия
	if(ms->memFd == -1) // в случае успеха: возврат неотрицательного дескриптора
		return -1;
	if(host)
		if(ftruncate(ms->memFd, MLTSHR_BLOCK_SIZE+sizeof(state))) // обрезка файла, определяемого fd , до указанного размера в байтах
			return -1;
	ms->buffer = mmap(NULL, MLTSHR_BLOCK_SIZE+sizeof(state), PROT_READ | PROT_WRITE, MAP_SHARED, ms->memFd, 0); // разделение использования отображения с другими процессами
	if(ms->buffer == (void*)-1)
		return -1;
	if(host)
	{
```

```
        pthread_mutexattr_t attrmutex;
        pthread_condattr_t attrcond;
        if(pthread_mutexattr_init(&attrmutex) ||                    // инициализация мьютекс
            pthread_mutexattr_setpshared(&attrmutex, PTHREAD_PROCESS_SHARED) ||  // позволить использовать мьютекс для синхронизации потоков в разных процессах.
            pthread_mutex_init(&(((state*)(ms->buffer))->mutex), &attrmutex) ||
            pthread_mutex_init(&(((state*)(ms->buffer))->write_mutex), &attrmutex) ||
            pthread_mutex_lock(&(((state*)(ms->buffer))->write_mutex)) ||
            pthread_condattr_init(&attrcond) ||                     // инициализация кондов
            pthread_condattr_setpshared(&attrcond, PTHREAD_PROCESS_SHARED) ||
            pthread_cond_init(&(((state*)(ms->buffer))->cond), &attrcond))
        return -1;
    }
    memcpy(ms->memname, memname, strlen(memname) + 1); // копируем кол-во байт из участка памяти на который дейсвует указатель, и +1 к кол-во символов
    return 0;
}

void mltshr_destroy(mltshr *ms)
{
    pthread_mutex_destroy(&(((state*)(ms->buffer))->mutex));  // После использования мьютекса его необходимо уничтожить
    pthread_mutex_destroy(&(((state*)(ms->buffer))->write_mutex));
    pthread_cond_destroy(&(((state*)(ms->buffer))->cond));  // используется для удаления переменной
    munmap(ms->buffer, MLTSHR_BLOCK_SIZE); // отображает длину в байтах
    shm_unlink(ms->memname); // удаляется имя объекта разделяемой памяти и, как только все процессы завершили работу с объектом и отменили его распределение, очищают пространство и уничтожают связанную с ним область памяти.
    close(ms->memFd);
    ms->memFd = -1;
```

```
}

void mltshr_write(mltshr *ms, char *msg, int mlen)
{
	pthread_mutex_lock(&(((state*)(ms->buffer))->write_mutex));
	pthread_mutex_lock(&(((state*)(ms->buffer))->mutex));
	((state*)(ms->buffer))->msglen = mlen;
	memcpy(ms->buffer+sizeof(state), msg, mlen);
	pthread_cond_broadcast(&(((state*)(ms->buffer))->cond)); // разблокировать все потоки, заблокированные в данный момент для указанной переменной условия cond
   pthread_mutex_unlock(&(((state*)(ms->buffer))->mutex));  // , заблокировавший мьютекс
}

char* mltshr_read(mltshr *ms, int *len)
{
	pthread_mutex_lock(&(((state*)(ms->buffer))->mutex)); // блокировка мьтекса
	pthread_mutex_unlock(&(((state*)(ms->buffer))->write_mutex)); // разюлокировка
	pthread_cond_wait(&(((state*)(ms->buffer))->cond), &(((state*)(ms->buffer))->mutex)); // возвращает запертый мьютекс, который принадлежит вызывающему потоку, даже если возникла ошибка. .
	int mlen = (((state*)(ms->buffer))->msglen);
	*len = mlen;
	char *mem = malloc(mlen);
	memcpy(mem, ms->buffer+sizeof(state), mlen);
	pthread_mutex_unlock(&(((state*)(ms->buffer))->mutex));
	return mem;
}

#endif
```

parent.c:

```
#include "unistd.h"
#include "stdio.h"
#include <fcntl.h>
#include <string.h>
#include "mltshr.h"

int createChild(char *fname)
{
	switch(fork())
	{
		case -1:
		{
			return -1;
		}
		case 0:
		{
			char *args[] = {NULL};
			execv(fname, args);
			return -1;
		}
		default:
		{
			break;
		}
	}
	return 0;
}

int main()
```

```
{
	mltshr parent_child1;
	mltshr child1_child2;
	mltshr child2_parent;
	if(mltshr_create(&parent_child1, "parent_child1", 1) || mltshr_create(&child1_child2, "child1_child2", 1) || mltshr_create(&child2_parent, "child2_parent", 1))
	{
		printf("error: cannot create shared memory\n");
		return 1;
	}
	createChild("./child1");
	createChild("./child2");
	printf("Enter string:\n");
	char buffer[256];
	while(1)
	{
		fgets(buffer, 255, stdin);
		int slen = strlen(buffer);
		mltshr_write(&parent_child1, buffer, slen+1);
		char *input = mltshr_read(&child2_parent, &slen);
		printf("%s", input);
		free(input);
		fflush(stdout);
	}
	return 0;
}
```

child1.c:

```
#include "unistd.h"
#include "stdio.h"
```

```
#include <string.h>
#include <ctype.h>
#include "mltshr.h"

void toUpper(char *str)
{
	int slen = strlen(str);
	for(int i = 0; i < slen; i++)
		str[i] = toupper(str[i]);
}

int main()
{
	mltshr parent_child1;
	mltshr child1_child2;
	if(mltshr_create(&parent_child1, "parent_child1", 0) || mltshr_create(&child1_child2,
"child1_child2", 0))
	{
		printf("error: cannot connect to shared memory\n");
		return 1;
	}
	while(1)
	{
		int inputLen;
		char *input = mltshr_read(&parent_child1, &inputLen);
		toUpper(input);
		mltshr_write(&child1_child2, input, strlen(input)+1);
		free(input);
	}
```

```
	return 0;
}
```

child2.c:

```
#include "unistd.h"
#include "stdio.h"
#include <string.h>
#include "mltshr.h"

void replaceSpaces(char *str, int* len)
{
	for(int i = 0; i < *len-1; i++)
		if(str[i] == ' ')
			while(str[i+1] == ' ') {
				for(int j = i+1; j<*len-1; j++)
					str[j] = str[j+1];
				*len -= 1;
			}
}

int main()
{
	mltshr child1_child2;
	mltshr child2_parent;
	if(mltshr_create(&child1_child2, "child1_child2", 0) || mltshr_create(&child2_parent, "child2_parent", 0))
	{
		printf("error: cannot connect to shared memory\n");
		return 1;
	}
```

```
        while(1)
        {
                int inputLen;
                char *input = mltshr_read(&child1_child2, &inputLen);
                replaceSpaces(input, &inputLen);
                mltshr_write(&child2_parent, input, strlen(input)+1);
                free(input);
        }
        return 0;
}
```

STRACE

```
execve("./parent", ["./parent"], 0x7fffea02ccf8 /* 51 vars */) = 0
brk(NULL)                               = 0x5559f4eec000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc5bef10f0) = -1 EINVAL (Недопустимый аргумент)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=77314, ...}) = 0
mmap(NULL, 77314, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fc397218000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/librt.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 7\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=40040, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fc397216000
mmap(NULL, 44000, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fc39720b000
mprotect(0x7fc39720e000, 24576, PROT_NONE) = 0
mmap(0x7fc39720e000, 16384, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7fc39720e000
mmap(0x7fc397212000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x7000) = 0x7fc397212000
mmap(0x7fc397214000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8000) = 0x7fc397214000
```

close(3)   = 0

openat(AT_FDCWD, "/lib/x86_64-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>\0\1\0\0\0\220\201\0\0\0\0\0\0"..., 832) = 832

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824) = 68

fstat(3, {st_mode=S_IFREG|0755, st_size=157224, ...}) = 0

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824) = 68

mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fc3971e8000

mmap(0x7fc3971ef000, 69632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0x7fc3971ef000

mmap(0x7fc397200000, 20480, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000) = 0x7fc397200000

mmap(0x7fc397205000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c000) = 0x7fc397205000

mmap(0x7fc397207000, 13432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fc397207000

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\360q\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\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68

fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 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

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\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68

mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fc396ff6000

mprotect(0x7fc39701b000, 1847296, PROT_NONE) = 0

mmap(0x7fc39701b000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7fc39701b000

mmap(0x7fc397193000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7fc397193000

mmap(0x7fc3971de000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7fc3971de000

mmap(0x7fc3971e4000, 13528, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fc3971e4000

close(3) = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fc396ff3000

arch_prctl(ARCH_SET_FS, 0x7fc396ff3740) = 0

mprotect(0x7fc3971de000, 12288, PROT_READ) = 0

mprotect(0x7fc397205000, 4096, PROT_READ) = 0

mprotect(0x7fc397214000, 4096, PROT_READ) = 0

mprotect(0x5559f3a08000, 4096, PROT_READ) = 0

mprotect(0x7fc397258000, 4096, PROT_READ) = 0

munmap(0x7fc397218000, 77314) = 0

set_tid_address(0x7fc396ff3a10) = 34267

set_robust_list(0x7fc396ff3a20, 24) = 0

rt_sigaction(SIGRTMIN, {sa_handler=0x7fc3971efbf0, sa_mask=[],
sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7fc3971fd3c0}, NULL, 8) = 0

rt_sigaction(SIGRT_1, {sa_handler=0x7fc3971efc90, sa_mask=[],
sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO, sa_restorer=0x7fc3971fd3c0}, 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

statfs("/dev/shm/", {f_type=TMPFS_MAGIC, f_bsize=4096, f_blocks=422081, f_bfree=413130,
f_bavail=413130, f_files=422081, f_ffree=421971, f_fsid={val=[0, 0]}, f_namelen=255, f_frsize=4096,
f_flags=ST_VALID|ST_NOSUID|ST_NODEV}) = 0

futex(0x7fc39720a390, FUTEX_WAKE_PRIVATE, 2147483647) = 0

openat(AT_FDCWD, "/dev/shm/parent_child1", O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC,
0600) = 3

ftruncate(3, 4232) = 0

mmap(NULL, 4232, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7fc397229000

openat(AT_FDCWD, "/dev/shm/child1_child2", O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC,
0600) = 4

ftruncate(4, 4232) = 0

mmap(NULL, 4232, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7fc397227000

openat(AT_FDCWD, "/dev/shm/child2_parent", O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0600) = 5

ftruncate(5, 4232)   = 0

mmap(NULL, 4232, PROT_READ|PROT_WRITE, MAP_SHARED, 5, 0) = 0x7fc397225000

clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLDstrace: Process 34268 attached

, child_tidptr=0x7fc396ff3a10) = 34268

[pid 34268] set_robust_list(0x7fc396ff3a20, 24 <unfinished ...>

[pid 34267] clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD <unfinished ...>

[pid 34268] <... set_robust_list resumed>) = 0

[pid 34268] execve("./child1", [], 0x7ffc5bef11d8 /* 51 vars */ <unfinished ...>

[pid 34267] <... clone resumed>, child_tidptr=0x7fc396ff3a10) = 34269

strace: Process 34269 attached

[pid 34267] fstat(1, <unfinished ...>

[pid 34269] set_robust_list(0x7fc396ff3a20, 24 <unfinished ...>

[pid 34267] <... fstat resumed>{st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0

[pid 34269] <... set_robust_list resumed>) = 0

[pid 34269] execve("./child2", [], 0x7ffc5bef11d8 /* 51 vars */ <unfinished ...>

[pid 34267] brk(NULL)   = 0x5559f4eec000

[pid 34268] <... execve resumed>)   = 0

[pid 34267] brk(0x5559f4f0d000)   = 0x5559f4f0d000

[pid 34268] brk(NULL <unfinished ...>

[pid 34267] write(1, "Enter string:\n", 14 <unfinished ...>

[pid 34268] <... brk resumed>)   = 0x560728f70000

[pid 34267] <... write resumed>)   = 14

[pid 34269] <... execve resumed>)   = 0

[pid 34267] fstat(0, <unfinished ...>

[pid 34268] arch_prctl(0x3001 /* ARCH_??? */, 0x7ffdef8897f0 <unfinished ...>

[pid 34267] <... fstat resumed>{st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0

[pid 34269] brk(NULL <unfinished ...>

[pid 34268] <... arch_prctl resumed>)   = -1 EINVAL (Недопустимый аргумент)

[pid 34267] read(0, <unfinished ...>

[pid 34269] <... brk resumed>)   = 0x5586d543d000

[pid 34268] access("/etc/ld.so.preload", R_OK <unfinished ...>

[pid 34269] arch_prctl(0x3001 /* ARCH_??? */, 0x7fff9d301aa0 <unfinished ...>

[pid 34268] <... access resumed>)   = -1 ENOENT (Нет такого файла или каталога)

[pid 34269] <... arch_prctl resumed>)   = -1 EINVAL (Недопустимый аргумент)

[pid 34268] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>

[pid 34269] access("/etc/ld.so.preload", R_OK <unfinished ...>

[pid 34268] <... openat resumed>)   = 3

[pid 34269] <... access resumed>)   = -1 ENOENT (Нет такого файла или каталога)

[pid 34268] fstat(3, <unfinished ...>

[pid 34269] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>

[pid 34268] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=77314, ...}) = 0

[pid 34268] mmap(NULL, 77314, PROT_READ, MAP_PRIVATE, 3, 0 <unfinished ...>

[pid 34269] <... openat resumed>)   = 3

[pid 34268] <... mmap resumed>)   = 0x7f86fd7cb000

[pid 34269] fstat(3, <unfinished ...>

[pid 34268] close(3 <unfinished ...>

[pid 34269] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=77314, ...}) = 0

[pid 34268] <... close resumed>)   = 0

[pid 34269] mmap(NULL, 77314, PROT_READ, MAP_PRIVATE, 3, 0 <unfinished ...>

[pid 34268] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/librt.so.1", O_RDONLY|O_CLOEXEC
<unfinished ...>

[pid 34269] <... mmap resumed>)   = 0x7f1d668cb000

[pid 34268] <... openat resumed>)   = 3

[pid 34269] close(3 <unfinished ...>

[pid 34268] read(3, <unfinished ...>

[pid 34269] <... close resumed>)   = 0

[pid 34268] <... read resumed>"\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 7\0\0\0\0\0\0"..., 832) =
832

[pid 34269] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/librt.so.1", O_RDONLY|O_CLOEXEC
<unfinished ...>

[pid 34268] fstat(3, <unfinished ...>

[pid 34269] <... openat resumed>)   = 3

[pid 34268] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=40040, ...}) = 0

[pid 34269] read(3, <unfinished ...>

[pid 34268] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 34269] <... read resumed>"\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0 7\0\0\0\0\0\0"..., 832) = 832

[pid 34268] <... mmap resumed>)   = 0x7f86fd7c9000

[pid 34269] fstat(3, {st_mode=S_IFREG|0644, st_size=40040, ...}) = 0

[pid 34268] mmap(NULL, 44000, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0 <unfinished ...>

[pid 34269] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 34268] <... mmap resumed>)   = 0x7f86fd7be000

[pid 34269] <... mmap resumed>)   = 0x7f1d668c9000

[pid 34268] mprotect(0x7f86fd7c1000, 24576, PROT_NONE <unfinished ...>

[pid 34269] mmap(NULL, 44000, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0 <unfinished ...>

[pid 34268] <... mprotect resumed>)   = 0

[pid 34269] <... mmap resumed>)   = 0x7f1d668be000

[pid 34268] mmap(0x7f86fd7c1000, 16384, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000 <unfinished ...>

[pid 34269] mprotect(0x7f1d668c1000, 24576, PROT_NONE <unfinished ...>

[pid 34268] <... mmap resumed>)   = 0x7f86fd7c1000

[pid 34269] <... mprotect resumed>)   = 0

[pid 34268] mmap(0x7f86fd7c5000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000 <unfinished ...>

[pid 34269] mmap(0x7f1d668c1000, 16384, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000 <unfinished ...>

[pid 34268] <... mmap resumed>)   = 0x7f86fd7c5000

[pid 34269] <... mmap resumed>)   = 0x7f1d668c1000

[pid 34268] mmap(0x7f86fd7c7000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8000 <unfinished ...>

[pid 34269] mmap(0x7f1d668c5000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000 <unfinished ...>

[pid 34268] <... mmap resumed>)   = 0x7f86fd7c7000

[pid 34269] <... mmap resumed>)   = 0x7f1d668c5000

[pid 34268] close(3 <unfinished ...>

[pid 34269] mmap(0x7f1d668c7000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8000 <unfinished ...>

[pid 34268] <... close resumed>)   = 0

[pid 34269] <... mmap resumed>)   = 0x7f1d668c7000

[pid 34269] close(3 <unfinished ...>

[pid 34268] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0", O_RDONLY|O_CLOEXEC
<unfinished ...>

[pid 34269] <... close resumed>)   = 0

[pid 34269] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0", O_RDONLY|O_CLOEXEC
<unfinished ...>

[pid 34268] <... openat resumed>)   = 3

[pid 34269] <... openat resumed>)   = 3

[pid 34268] read(3, <unfinished ...>

[pid 34269] read(3, <unfinished ...>

[pid 34268] <... read resumed>"\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\201\0\0\0\0\0\0"...,
832) = 832

[pid 34269] <... read resumed>"\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\201\0\0\0\0\0\0"...,
832) = 832

[pid 34268] pread64(3, <unfinished ...>

[pid 34269] pread64(3, <unfinished ...>

[pid 34268] <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824) = 68

[pid 34269] <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824) = 68

[pid 34268] fstat(3, <unfinished ...>

[pid 34269] fstat(3, <unfinished ...>

[pid 34268] <... fstat resumed>{st_mode=S_IFREG|0755, st_size=157224, ...}) = 0

[pid 34269] <... fstat resumed>{st_mode=S_IFREG|0755, st_size=157224, ...}) = 0

[pid 34268] pread64(3, <unfinished ...>

[pid 34269] pread64(3, <unfinished ...>

[pid 34268] <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824) = 68

[pid 34269] <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824) = 68

[pid 34268] mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0
<unfinished ...>

[pid 34269] mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0
<unfinished ...>

[pid 34268] <... mmap resumed>)   = 0x7f86fd79b000

[pid 34269] <... mmap resumed>)   = 0x7f1d6689b000

[pid 34269] mmap(0x7f1d668a2000, 69632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000 <unfinished ...>

[pid 34268] mmap(0x7f86fd7a2000, 69632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000 <unfinished ...>

[pid 34269] <... mmap resumed>)   = 0x7f1d668a2000

[pid 34268] <... mmap resumed>)   = 0x7f86fd7a2000

[pid 34269] mmap(0x7f1d668b3000, 20480, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000 <unfinished ...>

[pid 34268] mmap(0x7f86fd7b3000, 20480, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000 <unfinished ...>

[pid 34269] <... mmap resumed>)   = 0x7f1d668b3000

[pid 34268] <... mmap resumed>)   = 0x7f86fd7b3000

[pid 34269] mmap(0x7f1d668b8000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c000 <unfinished ...>

[pid 34268] mmap(0x7f86fd7b8000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c000 <unfinished ...>

[pid 34269] <... mmap resumed>)   = 0x7f1d668b8000

[pid 34268] <... mmap resumed>)   = 0x7f86fd7b8000

[pid 34269] mmap(0x7f1d668ba000, 13432, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 34268] mmap(0x7f86fd7ba000, 13432, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 34269] <... mmap resumed>)   = 0x7f1d668ba000

[pid 34268] <... mmap resumed>)   = 0x7f86fd7ba000

[pid 34269] close(3 <unfinished ...>

[pid 34268] close(3 <unfinished ...>

[pid 34269] <... close resumed>)   = 0

[pid 34268] <... close resumed>)   = 0

[pid 34269] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC
<unfinished ...>

[pid 34268] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC
<unfinished ...>

[pid 34269] <... openat resumed>)   = 3

[pid 34269] read(3, <unfinished ...>

[pid 34268] <... openat resumed>)   = 3

[pid 34269] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"..., 832)
= 832

[pid 34268] read(3, <unfinished ...>

[pid 34269] pread64(3, <unfinished ...>

[pid 34268] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"..., 832)
= 832

[pid 34269] <... pread64 resumed>"\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 34268] pread64(3, <unfinished ...>

[pid 34269] pread64(3, <unfinished ...>

[pid 34268] <... pread64 resumed>"\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 34269] <... pread64 resumed>"\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 34268] pread64(3, <unfinished ...>

[pid 34269] pread64(3, <unfinished ...>

[pid 34268] <... pread64 resumed>"\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 34269] <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"...,
68, 880) = 68

[pid 34268] pread64(3, <unfinished ...>

[pid 34269] fstat(3, <unfinished ...>

[pid 34268] <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"...,
68, 880) = 68

[pid 34269] <... fstat resumed>{st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0

[pid 34268] fstat(3, <unfinished ...>

[pid 34269] pread64(3, <unfinished ...>

[pid 34268] <... fstat resumed>{st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0

[pid 34269] <... pread64 resumed>"\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 34268] pread64(3, <unfinished ...>

[pid 34269] pread64(3, <unfinished ...>

[pid 34268] <... pread64 resumed>"\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 34269] <... pread64 resumed>"\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 34268] pread64(3, <unfinished ...>

[pid 34269] pread64(3, <unfinished ...>

[pid 34268] <... pread64 resumed>"\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 34269] <... pread64 resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68

[pid 34268] pread64(3, <unfinished ...>

[pid 34269] mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0 <unfinished ...>

[pid 34268] <... pread64 resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68, 880) = 68

[pid 34269] <... mmap resumed>)   = 0x7f1d666a9000

[pid 34268] mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0 <unfinished ...>

[pid 34269] mprotect(0x7f1d666ce000, 1847296, PROT_NONE <unfinished ...>

[pid 34268] <... mmap resumed>)   = 0x7f86fd5a9000

[pid 34269] <... mprotect resumed>)   = 0

[pid 34268] mprotect(0x7f86fd5ce000, 1847296, PROT_NONE <unfinished ...>

[pid 34269] mmap(0x7f1d666ce000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000 <unfinished ...>

[pid 34268] <... mprotect resumed>)   = 0

[pid 34269] <... mmap resumed>) = 0x7f1d666ce000

[pid 34268] mmap(0x7f86fd5ce000, 1540096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000 <unfinished ...>

[pid 34269] mmap(0x7f1d66846000, 303104, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000 <unfinished ...>

[pid 34268] <... mmap resumed>) = 0x7f86fd5ce000

[pid 34269] <... mmap resumed>) = 0x7f1d66846000

[pid 34268] mmap(0x7f86fd746000, 303104, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000 <unfinished ...>

[pid 34269] mmap(0x7f1d66891000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000 <unfinished ...>

[pid 34268] <... mmap resumed>) = 0x7f86fd746000

[pid 34269] <... mmap resumed>) = 0x7f1d66891000

[pid 34268] mmap(0x7f86fd791000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000 <unfinished ...>

[pid 34269] mmap(0x7f1d66897000, 13528, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 34268] <... mmap resumed>) = 0x7f86fd791000

[pid 34269] <... mmap resumed>) = 0x7f1d66897000

[pid 34268] mmap(0x7f86fd797000, 13528, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 34269] close(3 <unfinished ...>

[pid 34268] <... mmap resumed>) = 0x7f86fd797000

[pid 34269] <... close resumed>) = 0

[pid 34268] close(3) = 0

[pid 34269] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
-1, 0) = 0x7f1d666a6000

[pid 34268] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
-1, 0 <unfinished ...>

[pid 34269] arch_prctl(ARCH_SET_FS, 0x7f1d666a6740 <unfinished ...>

[pid 34268] <... mmap resumed>) = 0x7f86fd5a6000

[pid 34269] <... arch_prctl resumed>) = 0

[pid 34268] arch_prctl(ARCH_SET_FS, 0x7f86fd5a6740) = 0

[pid 34269] mprotect(0x7f1d66891000, 12288, PROT_READ <unfinished ...>

[pid 34268] mprotect(0x7f86fd791000, 12288, PROT_READ <unfinished ...>

[pid 34269] <... mprotect resumed>)   = 0

[pid 34268] <... mprotect resumed>)   = 0

[pid 34269] mprotect(0x7f1d668b8000, 4096, PROT_READ <unfinished ...>

[pid 34268] mprotect(0x7f86fd7b8000, 4096, PROT_READ <unfinished ...>

[pid 34269] <... mprotect resumed>)   = 0

[pid 34268] <... mprotect resumed>)   = 0

[pid 34269] mprotect(0x7f1d668c7000, 4096, PROT_READ <unfinished ...>

[pid 34268] mprotect(0x7f86fd7c7000, 4096, PROT_READ <unfinished ...>

[pid 34269] <... mprotect resumed>)   = 0

[pid 34268] <... mprotect resumed>)   = 0

[pid 34269] mprotect(0x5586d4cdd000, 4096, PROT_READ <unfinished ...>

[pid 34268] mprotect(0x560728858000, 4096, PROT_READ <unfinished ...>

[pid 34269] <... mprotect resumed>)   = 0

[pid 34268] <... mprotect resumed>)   = 0

[pid 34269] mprotect(0x7f1d6690b000, 4096, PROT_READ <unfinished ...>

[pid 34268] mprotect(0x7f86fd80b000, 4096, PROT_READ <unfinished ...>

[pid 34269] <... mprotect resumed>)   = 0

[pid 34268] <... mprotect resumed>)   = 0

[pid 34269] munmap(0x7f1d668cb000, 77314 <unfinished ...>

[pid 34268] munmap(0x7f86fd7cb000, 77314 <unfinished ...>

[pid 34269] <... munmap resumed>)   = 0

[pid 34268] <... munmap resumed>)   = 0

[pid 34269] set_tid_address(0x7f1d666a6a10 <unfinished ...>

[pid 34268] set_tid_address(0x7f86fd5a6a10 <unfinished ...>

[pid 34269] <... set_tid_address resumed>) = 34269

[pid 34268] <... set_tid_address resumed>) = 34268

[pid 34269] set_robust_list(0x7f1d666a6a20, 24 <unfinished ...>

[pid 34268] set_robust_list(0x7f86fd5a6a20, 24 <unfinished ...>

[pid 34269] <... set_robust_list resumed>) = 0

[pid 34268] <... set_robust_list resumed>) = 0

[pid 34269] rt_sigaction(SIGRTMIN, {sa_handler=0x7f1d668a2bf0, sa_mask=[],
sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7f1d668b03c0}, <unfinished ...>

[pid 34268] rt_sigaction(SIGRTMIN, {sa_handler=0x7f86fd7a2bf0, sa_mask=[],
sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7f86fd7b03c0}, <unfinished ...>

[pid 34269] <... rt_sigaction resumed>NULL, 8) = 0

[pid 34268] <... rt_sigaction resumed>NULL, 8) = 0

[pid 34269] rt_sigaction(SIGRT_1, {sa_handler=0x7f1d668a2c90, sa_mask=[],
sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f1d668b03c0}, <unfinished ...>

[pid 34268] rt_sigaction(SIGRT_1, {sa_handler=0x7f86fd7a2c90, sa_mask=[],
sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f86fd7b03c0}, <unfinished ...>

[pid 34269] <... rt_sigaction resumed>NULL, 8) = 0

[pid 34268] <... rt_sigaction resumed>NULL, 8) = 0

[pid 34269] rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], <unfinished ...>

[pid 34268] rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], <unfinished ...>

[pid 34269] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 34268] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 34269] prlimit64(0, RLIMIT_STACK, NULL, <unfinished ...>

[pid 34268] prlimit64(0, RLIMIT_STACK, NULL, <unfinished ...>

[pid 34269] <... prlimit64 resumed>{rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

[pid 34268] <... prlimit64 resumed>{rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

[pid 34269] statfs("/dev/shm/", <unfinished ...>

[pid 34268] statfs("/dev/shm/", <unfinished ...>

[pid 34269] <... statfs resumed>{f_type=TMPFS_MAGIC, f_bsize=4096, f_blocks=422081,
f_bfree=413127, f_bavail=413127, f_files=422081, f_ffree=421968, f_fsid={val=[0, 0]}, f_namelen=255,
f_frsize=4096, f_flags=ST_VALID|ST_NOSUID|ST_NODEV}) = 0

[pid 34269] futex(0x7f1d668bd390, FUTEX_WAKE_PRIVATE, 2147483647 <unfinished ...>

[pid 34268] <... statfs resumed>{f_type=TMPFS_MAGIC, f_bsize=4096, f_blocks=422081,
f_bfree=413127, f_bavail=413127, f_files=422081, f_ffree=421968, f_fsid={val=[0, 0]}, f_namelen=255,
f_frsize=4096, f_flags=ST_VALID|ST_NOSUID|ST_NODEV}) = 0

[pid 34269] <... futex resumed>) = 0

[pid 34268] futex(0x7f86fd7bd390, FUTEX_WAKE_PRIVATE, 2147483647) = 0

[pid 34269] openat(AT_FDCWD, "/dev/shm/child1_child2",
O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0600 <unfinished ...>

[pid 34268] openat(AT_FDCWD, "/dev/shm/parent_child1",
O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0600 <unfinished ...>

[pid 34269] <... openat resumed>)   = 3

[pid 34268] <... openat resumed>)   = 3

[pid 34269] mmap(NULL, 4232, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0 <unfinished ...>

[pid 34268] mmap(NULL, 4232, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0 <unfinished ...>

[pid 34269] <... mmap resumed>)   = 0x7f1d668dc000

[pid 34268] <... mmap resumed>)   = 0x7f86fd7dc000

[pid 34269] openat(AT_FDCWD, "/dev/shm/child2_parent",
O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0600 <unfinished ...>

[pid 34268] openat(AT_FDCWD, "/dev/shm/child1_child2",
O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0600 <unfinished ...>

[pid 34269] <... openat resumed>)   = 4

[pid 34268] <... openat resumed>)   = 4

[pid 34269] mmap(NULL, 4232, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0 <unfinished ...>

[pid 34268] mmap(NULL, 4232, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0 <unfinished ...>

[pid 34269] <... mmap resumed>)   = 0x7f1d668da000

[pid 34268] <... mmap resumed>)   = 0x7f86fd7da000

[pid 34269] futex(0x7f1d668dc078, FUTEX_WAIT, 0, NULL <unfinished ...>

[pid 34268] futex(0x7f86fd7dc078, FUTEX_WAIT, 0, NULL <unfinished ...>

[pid 34267] <... read resumed>"GBIGBI \n", 1024) = 8

[pid 34267] futex(0x7fc397229078, FUTEX_WAKE, 2147483647) = 1

[pid 34268] <... futex resumed>)   = 0

[pid 34267] futex(0x7fc397225078, FUTEX_WAIT, 0, NULL <unfinished ...>

[pid 34268] brk(NULL)   = 0x560728f70000

[pid 34268] brk(0x560728f91000)   = 0x560728f91000

[pid 34268] futex(0x7f86fd7dc000, FUTEX_WAKE, 1) = 0

[pid 34268] futex(0x7f86fd7da078, FUTEX_WAKE, 2147483647 <unfinished ...>

[pid 34269] <... futex resumed>)   = 0

[pid 34268] <... futex resumed>)   = 1

[pid 34269] futex(0x7f1d668dc000, FUTEX_WAIT, 2, NULL <unfinished ...>

[pid 34268] futex(0x7f86fd7da000, FUTEX_WAKE, 1 <unfinished ...>

[pid 34269] <... futex resumed>)   = -1 EAGAIN (Ресурс временно недоступен)

[pid 34268] <... futex resumed>)   = 0

[pid 34269] brk(NULL <unfinished ...>

[pid 34268] futex(0x7f86fd7dc07c, FUTEX_WAIT, 0, NULL <unfinished ...>

[pid 34269] <... brk resumed>)   = 0x5586d543d000

[pid 34269] brk(0x5586d545e000)   = 0x5586d545e000

[pid 34269] futex(0x7f1d668dc000, FUTEX_WAKE, 1) = 0

[pid 34269] futex(0x7f1d668da078, FUTEX_WAKE, 2147483647 <unfinished ...>

[pid 34267] <... futex resumed>)   = 0

[pid 34269] <... futex resumed>)   = 1

[pid 34267] futex(0x7fc397225000, FUTEX_WAIT, 2, NULL <unfinished ...>

[pid 34269] futex(0x7f1d668da000, FUTEX_WAKE, 1 <unfinished ...>

[pid 34267] <... futex resumed>)   = -1 EAGAIN (Ресурс временно недоступен)

[pid 34269] <... futex resumed>)   = 0

[pid 34267] futex(0x7fc397225000, FUTEX_WAKE, 1 <unfinished ...>

[pid 34269] futex(0x7f1d668dc07c, FUTEX_WAIT, 0, NULL <unfinished ...>

[pid 34267] <... futex resumed>)   = 0

[pid 34267] write(1, "gbigbi_\n", 8)   = 8

[pid 34267] read(0, 0x5559f4eec6b0, 1024) = ? ERESTARTSYS (To be restarted if SA_RESTART is set)

[pid 34267] read(0, 0x5559f4eec6b0, 1024) = ? ERESTARTSYS (To be restarted if SA_RESTART is set)

[pid 34269] <... futex resumed>)   = ? ERESTARTSYS (To be restarted if SA_RESTART is set)

[pid 34268] <... futex resumed>)   = ? ERESTARTSYS (To be restarted if SA_RESTART is set)

[pid 34267] --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---

[pid 34269] --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---

[pid 34268] --- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---

[pid 34267] read(0, <unfinished ...>

[pid 34269] futex(0x7f1d668dc07c, FUTEX_WAIT, 0, NULL <unfinished ...>

[pid 34268] futex(0x7f86fd7dc07c, FUTEX_WAIT, 0, NULL <unfinished ...>

[pid 34267] <... read resumed>0x5559f4eec6b0, 1024) = ? ERESTARTSYS (To be restarted if SA_RESTART is set)

strace: Process 34267 detached

strace: Process 34268 detached

strace: Process 34269 detached

Вывод

В процессе выполнения данной лабораторной работы я научился обеспечивать обмен данными между процессами посредством технологии «File mapping» и синхронизаци. До этого я умел пользоваться только не именованными пайпами, теперь могу обращаться к общей памяти по имени, чем-то похоже на именованный пайп, о существовании которого я знаю и буду использовать в курсовом проекте.