****Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работ №1 по курсу**

**«Операционные системы»**

Группа: М80 – 206Б-18

Студент: Касимов М.М.

Преподаватель: Соколов А.А.

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

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2019.

**Содержание**

1. Постановка задачи
2. Общие сведения о программе
3. Средство диагностики
4. Основные файлы программы
5. Демонстрация работы программы
6. Вывод

**Постановка задачи**.

Приобретение практических навыков диагностики работы программного обеспечения.

**Общие сведения о программе**

Программа компилируется из одного файла lab4.c. В данном файле используются заголовочные файлы stdio.h, unistd.h, stdbool.h, stdlib.h, wait.h, sys/types.h, semaphore.h, fcntl.h. В программе используются следующие вызовы:

1. **mkstemp** – для создания временного файла
2. **sem\_open** – для создания нового именованного семафора или открытия уже существующего.
3. **sem\_unlink** – для удаления именованного семафора.
4. **fork** – для создания дочернего процесса.
5. **sem\_post** – для увеличения(разблокировки) семафора.
6. **sem\_wait –** для уменьшения(блокировки) семафора.
7. **mmap –** для отображения файла в адресное пространство процесса.

**Средство диагностики**

Утилита **strace**.

**Основные файлы программы**.

**Файл lab4.c**

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <wait.h>

#include <stdlib.h>

#include <stdbool.h>

#include <sys/mman.h>

#include <fcntl.h>

#include <semaphore.h>

#include <sys/stat.h>

#include <string.h>

#define BUFFER\_SIZE 10

//создание временного файла.

int create\_tmp() {

char tmp\_name[] = "/tmp/tmpf.XXXXXX";

int tmp\_fd = mkstemp(tmp\_name);

if ( tmp\_fd == -1) {

printf("error\n");

exit(1);

}

int size = BUFFER\_SIZE + 1;

char array[size];

for ( int i = 0; i < size; ++i ) {

array[i] = '\0';

}

write(tmp\_fd, array, size);

return tmp\_fd;

}

//рекурсивное вычисление факториала.

//создаём семафор и заставляем родительский процесс ждать пока дочерний не завершит свои вычисления.

unsigned long long fact(int n, int\* map){

if (n == 0){

return 1;

}

else {

const char\* out\_sem\_name = "/o\_s";

sem\_unlink(out\_sem\_name);

sem\_t\* out = sem\_open(out\_sem\_name, O\_CREAT, 777, 0);

pid\_t proc = fork();

if (proc < 0){

printf("Error: fork\n");

exit(1);

}

if (proc == 0){ //дочерний процесс

unsigned long long res;

res = fact(n - 1, map);

map[0] = res;

sem\_post(out);

exit(0);

}

if (proc > 0) { //родительский процесс

sem\_wait(out);

unsigned long long res;

res = map[0];

return n \* (res);

}

}

}

int main(){

char a[132] = "Instruction.\n Enter only one nonnegative integer number less than 14. In case of incorrect input, the program will simply terminate:";

for ( int i = 0 ; i < 132 ; ++i ) {

write(STDOUT\_FILENO,&a[i],sizeof(char));

} //вывод короткой инструкции.

int flag = 0,flagPlus = 0,flagTabs = 0,flagNumber = 0;

int n = 0;

char c;

while(true) {

read(STDIN\_FILENO,&c,1);

if (c <= '9' && c >= '0') {

flagPlus++;

flagNumber++;

n \*= 10;

n += c - '0';

continue;

}

if (c=='\n')

break;

if (c == '+' && flagPlus == 0) {

flagPlus++;

continue;

}

if ((c == ' ' || c == '\t') && (flagTabs == 0)) {

continue;

}

else

++flag;

}

if (flag != 0 || flagNumber == 0) {

return 0;

}

if (n > 13)

return 0; // парсер.

unsigned long long k;

int fd = create\_tmp();// дескриптор временного файла.

//mapping файла.

int\* map = (int\*) mmap(NULL,10,PROT\_WRITE | PROT\_READ, MAP\_SHARED, fd, 0);

if (map == NULL) {

printf("error mapping\n");

exit(1);

}

k=fact(n,map);

printf("result %lld\n",k);

return 0;

}

**Демонстрация работы программы.**

**Запустим утилиту strace для 4 лабораторной работы.**

execve("./a.out", ["./a.out"], 0x7fffcf121a50 /\* 18 vars \*/) = 0 brk(NULL) = 0x7fffe1458000 access("/etc/ld.so.nohwcap", F\_OK) = -1 ENOENT (No such file or directory) 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=26962, ...}) = 0 mmap(NULL, 26962, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f8305f6d000 close(3) = 0 access("/etc/ld.so.nohwcap", F\_OK) = -1 ENOENT (No such file or directory) 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\0000b\0\0\0\0\0\0"..., 832) = 832 fstat(3, {st\_mode=S\_IFREG|0755, st\_size=144976, ...}) = 0 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f8305f60000 mmap(NULL, 2221184, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f83059e0000 mprotect(0x7f83059fa000, 2093056, PROT\_NONE) = 0 mmap(0x7f8305bf9000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19000) = 0x7f8305bf9000 mmap(0x7f8305bfb000, 13440, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f8305bfb000 close(3) = 0 access("/etc/ld.so.nohwcap", F\_OK) = -1 ENOENT (No such file or directory) 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\260\34\2\0\0\0\0\0"..., 832) = 832 fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2030544, ...}) = 0 mmap(NULL, 4131552, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f83055e0000 mprotect(0x7f83057c7000, 2097152, PROT\_NONE) = 0 mmap(0x7f83059c7000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7f83059c7000 mmap(0x7f83059cd000, 15072, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f83059cd000 close(3) = 0 mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f8305f50000 arch\_prctl(ARCH\_SET\_FS, 0x7f8305f50740) = 0 mprotect(0x7f83059c7000, 16384, PROT\_READ) = 0 mprotect(0x7f8305bf9000, 4096, PROT\_READ) = 0 mprotect(0x7f8306201000, 4096, PROT\_READ) = 0 mprotect(0x7f8305e27000, 4096, PROT\_READ) = 0 munmap(0x7f8305f6d000, 26962) = 0 set\_tid\_address(0x7f8305f50a10) = 473 set\_robust\_list(0x7f8305f50a20, 24) = 0 rt\_sigaction(SIGRTMIN, {sa\_handler=0x7f83059e5cb0, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_SIGINFO, sa\_restorer=0x7f83059f2890}, NULL, 8) = 0 rt\_sigaction(SIGRT\_1, {sa\_handler=0x7f83059e5d50, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7f83059f2890}, NULL, 8) = 0 rt\_sigprocmask(SIG\_UNBLOCK, [RTMIN RT\_1], NULL, 8) = 0 prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=8192\*1024}) = 0 write(1, "I", 1I) = 1

…

write(1, ":", 1:) = 1 read(0, 3 "3", 1) = 1 read(0, "\n", 1) = 1 gettimeofday({tv\_sec=1577352060, tv\_usec=521412}, NULL) = 0 getpid() = 473 openat(AT\_FDCWD, "/tmp/tmpf.PyOENI", O\_RDWR|O\_CREAT|O\_EXCL, 0600) = 3 write(3, "\0\0\0\0\0\0\0\0\0\0\0", 11) = 11 mmap(NULL, 10, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7f8305f73000 statfs("/dev/shm/", {f\_type=TMPFS\_MAGIC, f\_bsize=4096, f\_blocks=31123199, f\_bfree=10038021, f\_bavail=10038021, f\_files=999, f\_ffree=1000000, f\_fsid={val=[1, 0]}, f\_namelen=255, f\_frsize=4096, f\_flags=ST\_VALID|ST\_NOSUID|ST\_NODEV|ST\_NOATIME}) = 0 futex(0x7f8305bfe370, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0 unlink("/dev/shm/sem.o\_s") = 0 openat(AT\_FDCWD, "/dev/shm/sem.o\_s", O\_RDWR|O\_NOFOLLOW) = -1 ENOENT (No such file or directory) getpid() = 473 lstat("/dev/shm/tkEQwH", 0x7fffe8cf0540) = -1 ENOENT (No such file or directory) openat(AT\_FDCWD, "/dev/shm/tkEQwH", O\_RDWR|O\_CREAT|O\_EXCL, 01411) = 4 write(4, "\0\0\0\0\0\0\0\0\200\0\0\0\203\177\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, 4, 0) = 0x7f8305f72000 link("/dev/shm/tkEQwH", "/dev/shm/sem.o\_s") = 0 fstat(4, {st\_mode=S\_IFREG|S\_ISVTX|0411, st\_size=32, ...}) = 0 brk(NULL) = 0x7fffe1458000 brk(0x7fffe1479000) = 0x7fffe1479000 unlink("/dev/shm/tkEQwH") = 0 close(4) = 0 clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f8305f50a10) = 474 futex(0x7f8305f72000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, 0xffffffff) = 0 --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=474, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} --- fstat(1, {st\_mode=S\_IFCHR|0660, st\_rdev=makedev(4, 1), ...}) = 0 ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0 write(1, "result 6\n", 9result 6 ) = 9 exit\_group(0) = ? +++ exited with 0 +++

**Теперь запустим с ключом –f , который позволяет следить за дочерними процессами(потоками).**

write(1, ":", 1:) = 1 read(0, 3 "3", 1) = 1 read(0, "\n", 1) = 1 gettimeofday({tv\_sec=1577352133, tv\_usec=40820}, NULL) = 0 getpid() = 479 openat(AT\_FDCWD, "/tmp/tmpf.Ofaxng", O\_RDWR|O\_CREAT|O\_EXCL, 0600) = 3 write(3, "\0\0\0\0\0\0\0\0\0\0\0", 11) = 11 mmap(NULL, 10, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7fc4e63d7000 statfs("/dev/shm/", {f\_type=TMPFS\_MAGIC, f\_bsize=4096, f\_blocks=31123199, f\_bfree=10038256, f\_bavail=10038256, f\_files=999, f\_ffree=1000000, f\_fsid={val=[1, 0]}, f\_namelen=255, f\_frsize=4096, f\_flags=ST\_VALID|ST\_NOSUID|ST\_NODEV|ST\_NOATIME}) = 0 futex(0x7fc4e5ffe370, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0 unlink("/dev/shm/sem.o\_s") = 0 openat(AT\_FDCWD, "/dev/shm/sem.o\_s", O\_RDWR|O\_NOFOLLOW) = -1 ENOENT (No such file or directory) getpid() = 479 lstat("/dev/shm/vNeT5i", 0x7fffcbeca090) = -1 ENOENT (No such file or directory) openat(AT\_FDCWD, "/dev/shm/vNeT5i", O\_RDWR|O\_CREAT|O\_EXCL, 01411) = 4 write(4, "\0\0\0\0\0\0\0\0\200\0\0\0\304\177\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, 4, 0) = 0x7fc4e63d6000 link("/dev/shm/vNeT5i", "/dev/shm/sem.o\_s") = 0 fstat(4, {st\_mode=S\_IFREG|S\_ISVTX|0411, st\_size=32, ...}) = 0 brk(NULL) = 0x7fffc4eff000 brk(0x7fffc4f20000) = 0x7fffc4f20000 unlink("/dev/shm/vNeT5i") = 0 close(4) = 0 clone(strace: Process 480 attached child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7fc4e63b0a10) = 480 [pid 480] set\_robust\_list(0x7fc4e63b0a20, 24 <unfinished ...> [pid 479] futex(0x7fc4e63d6000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, 0xffffffff <unfinished ...> [pid 480] <... set\_robust\_list resumed> ) = 0 [pid 480] unlink("/dev/shm/sem.o\_s") = 0 [pid 480] openat(AT\_FDCWD, "/dev/shm/sem.o\_s", O\_RDWR|O\_NOFOLLOW) = -1 ENOENT (No such file or directory) [pid 480] getpid() = 480 [pid 480] lstat("/dev/shm/F83FUl", 0x7fffcbeca040) = -1 ENOENT (No such file or directory) [pid 480] openat(AT\_FDCWD, "/dev/shm/F83FUl", O\_RDWR|O\_CREAT|O\_EXCL, 01411) = 4 [pid 480] write(4, "\0\0\0\0\0\0\0\0\200\0\0\0\377\377\377\377\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 32) = 32 [pid 480] mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 4, 0) = 0x7fc4e63d5000 [pid 480] link("/dev/shm/F83FUl", "/dev/shm/sem.o\_s") = 0 [pid 480] fstat(4, {st\_mode=S\_IFREG|S\_ISVTX|0411, st\_size=32, ...}) = 0 [pid 480] unlink("/dev/shm/F83FUl") = 0 [pid 480] close(4) = 0 [pid 480] clone(strace: Process 481 attached child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7fc4e63b0a10) = 481 [pid 481] set\_robust\_list(0x7fc4e63b0a20, 24 <unfinished ...> [pid 480] futex(0x7fc4e63d5000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, 0xffffffff <unfinished ...> [pid 481] <... set\_robust\_list resumed> ) = 0 [pid 481] unlink("/dev/shm/sem.o\_s") = 0 [pid 481] openat(AT\_FDCWD, "/dev/shm/sem.o\_s", O\_RDWR|O\_NOFOLLOW) = -1 ENOENT (No such file or directory) [pid 481] getpid() = 481 [pid 481] lstat("/dev/shm/z5PkPo", 0x7fffcbec9ff0) = -1 ENOENT (No such file or directory) [pid 481] openat(AT\_FDCWD, "/dev/shm/z5PkPo", O\_RDWR|O\_CREAT|O\_EXCL, 01411) = 4 [pid 481] write(4, "\0\0\0\0\0\0\0\0\200\0\0\0\377\377\377\377\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 32) = 32 [pid 481] mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 4, 0) = 0x7fc4e63d4000 [pid 481] link("/dev/shm/z5PkPo", "/dev/shm/sem.o\_s") = 0 [pid 481] fstat(4, {st\_mode=S\_IFREG|S\_ISVTX|0411, st\_size=32, ...}) = 0 [pid 481] unlink("/dev/shm/z5PkPo") = 0 [pid 481] close(4) = 0 [pid 481] clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7fc4e63b0a10) = 482 strace: Process 482 attached [pid 481] futex(0x7fc4e63d4000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, 0xffffffff <unfinished ...> [pid 482] set\_robust\_list(0x7fc4e63b0a20, 24) = 0 [pid 482] futex(0x7fc4e63d4000, FUTEX\_WAKE, 1) = 1 [pid 481] <... futex resumed> ) = 0 [pid 482] exit\_group(0 <unfinished ...> [pid 481] futex(0x7fc4e63d5000, FUTEX\_WAKE, 1) = 1 [pid 480] <... futex resumed> ) = 0 [pid 481] exit\_group(0 <unfinished ...> [pid 480] futex(0x7fc4e63d6000, FUTEX\_WAKE, 1) = 1 [pid 479] <... futex resumed> ) = 0 [pid 480] exit\_group(0 <unfinished ...> [pid 479] fstat(1, <unfinished ...> [pid 480] <... exit\_group resumed>) = ? [pid 479] <... fstat resumed> {st\_mode=S\_IFCHR|0660, st\_rdev=makedev(4, 1), ...}) = 0 [pid 480] +++ exited with 0 +++ [pid 479] --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=480, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} --- [pid 479] ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0 [pid 479] write(1, "result 6\n", 9result 6 <unfinished ...> [pid 481] <... exit\_group resumed>) = ? [pid 479] <... write resumed> ) = 9 [pid 481] +++ exited with 0 +++ [pid 479] exit\_group(0) = ? [pid 479] +++ exited with 0 +++ <... exit\_group resumed>) = ? +++ exited with 0 +++

**Так же можно использовать ключ –T и выводить длительность сис-го вызова.**

**Вывод.**

Я научился наблюдать за системными вызовами в Unix, используя утилиту strace. Данная утилита имеет много ключей, тем самым она является гибким инструментом для нахождения ошибок, связанных с системными вызовами.