

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

**Лабораторная работа №1 по курсу
«Операционные системы»**

Студент: Пивницкий Д.С.
Группа: М8о–206Б–19

Преподаватель: Соколов Андрей Алексеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2020.

Постановка задачи

Цель работы

Приобретение практических навыков в:

- Диагностики работы программного обеспечения.

Задание

При выполнении последующих лабораторных работ необходимо продемонстрировать ключевые системные вызовы, которые в них используются и то, что их использование соответствует варианту ЛР

Описание

Для выполнения демонстрации strace была использована скомпилированная 4 лабораторная работа.

Пример работы

(py37) → ~ cd os

cd: Нет такого файла или каталога: os

(py37) → ~ cd OS

(py37) → OS cd oslab4

(py37) → oslab4 ls

lab4.c out res.txt test.txt

(py37) → oslab4 strace ./out

execve("./out", ["./out"], 0x7ffde341f220 /* 61 vars */) = 0

brk(NULL) = 0x55f16fa49000

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=118856, ...}) = 0

mmap(NULL, 118856, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f7ab7602000

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\3\0>\0\1\0\0\0000b\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) = 0x7f7ab7600000

mmap(NULL, 2221184, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f7ab71d8000

mprotect(0x7f7ab71f2000, 2093056, PROT_NONE) = 0

mmap(0x7f7ab73f1000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19000) = 0x7f7ab73f1000

mmap(0x7f7ab73f3000, 13440, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f7ab73f3000

```

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\3\0>\0\1\0\0\0\20\35\2\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0755, st_size=2030928, ...}) = 0

mmap(NULL, 4131552, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f7ab6de7000

mprotect(0x7f7ab6fce000, 2097152, PROT_NONE) = 0

mmap(0x7f7ab71ce000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1e7000) = 0x7f7ab71ce000

mmap(0x7f7ab71d4000, 15072, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7f7ab71d4000

close(3)                = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f7ab75fd000

arch_prctl(ARCH_SET_FS, 0x7f7ab75fd740) = 0

mprotect(0x7f7ab71ce000, 16384, PROT_READ) = 0

mprotect(0x7f7ab73f1000, 4096, PROT_READ) = 0

mprotect(0x55f16da87000, 4096, PROT_READ) = 0

mprotect(0x7f7ab7620000, 4096, PROT_READ) = 0

munmap(0x7f7ab7602000, 118856)        = 0

set_tid_address(0x7f7ab75fda10)      = 13230

set_robust_list(0x7f7ab75fda20, 24)  = 0

rt_sigaction(SIGRTMIN, {sa_handler=0x7f7ab71ddcb0, sa_mask=[], sa_flags=SA_RESTORER|SA_SIGINFO,
sa_restorer=0x7f7ab71ea980}, NULL, 8) = 0

rt_sigaction(SIGRT_1, {sa_handler=0x7f7ab71ddd50, sa_mask=[], sa_flags=SA_RESTORER|SA_RESTART|
SA_SIGINFO, sa_restorer=0x7f7ab71ea980}, 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

fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0

brk(NULL)                  = 0x55f16fa49000

brk(0x55f16fa6a000)        = 0x55f16fa6a000

write(1, "INVALID COUNT OF ARGS\n", 22INVALID COUNT OF ARGS
) = 22

```

```
write(1, "USAGE: ./out <file>\n", 20USAGE: ./out <file>
```

```
) = 20
```

```
exit_group(-1)          = ?
```

```
+++ exited with 255 +++
```

Вывод

Существуют специальные системные вызовы(`fork`) для создания процессов, также существуют специальные каналы `pipe`, которые позволяют связать процессы и обмениваться данными при помощи этих `pipe`-ов. При использовании `fork` важно помнить, что фактически создается копия вашего текущего процесса и неправильная работа может привести к неожиданным результатам и последствиям, однако создание процессов очень удобно, когда вам нужно выполнять несколько действий параллельно. Также у каждого процесса есть свой `id`, по которому его можно определить. Также важно работать с чтением и записью из канала, помня что `read`, `write` возвращает количество успешно считанных/записанных байт и оно не обязательно равно тому значению, которое вы указали. Также важно не забывать закрывать `pipe` после завершения работы.