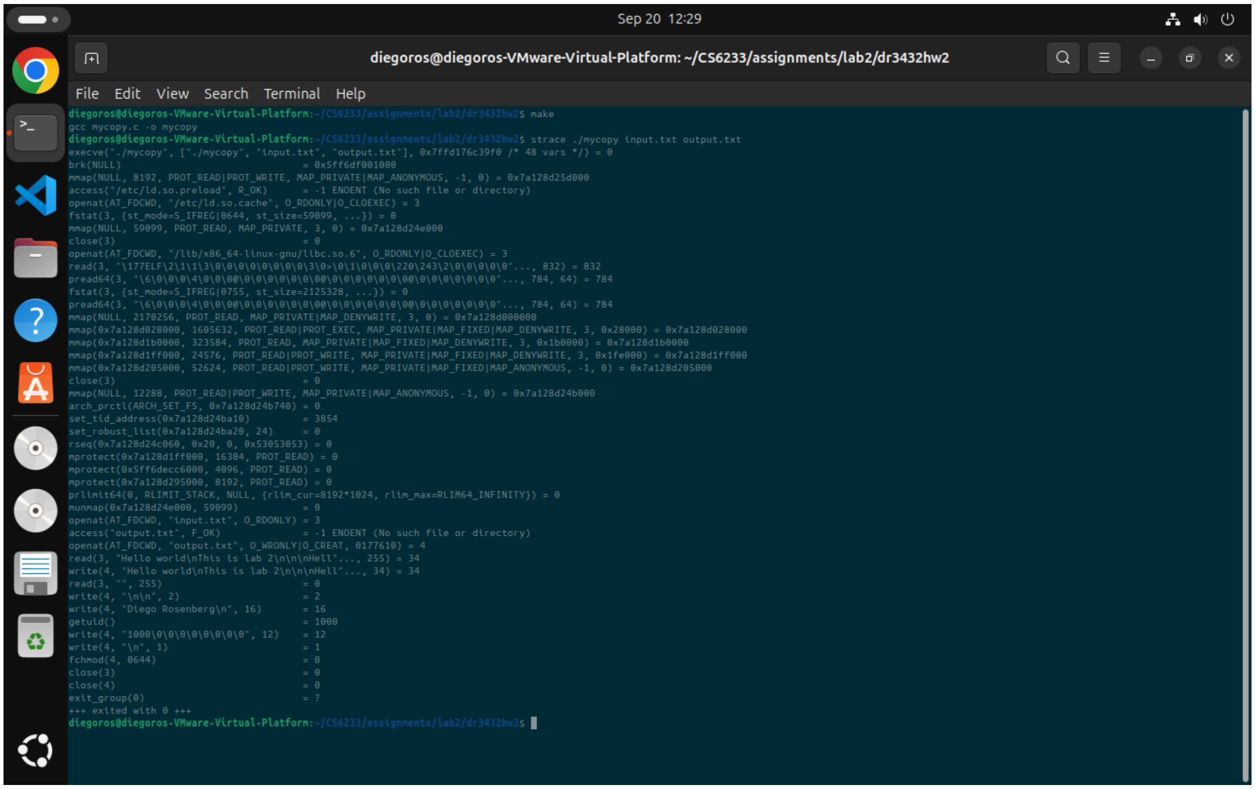
Lab2 Results

Student: Diego Rosenberg de Angoitia

Net ID: dr3432

N#: N11409122



Sep 20 12:29
diegoros@diegoros-VMware-Virtual-Platform: ~/CS6233/assignments/lab2/dr3432hw2
File
Edit
View
Search
Terminal
Help

```
diegoros@diegoros-VMware-Virtual-Platform:~/CS6233/assignments/lab2/dr3432hw2$ make
gcc mycopy.c -o mycopy
diegoros@diegoros-VMware-Virtual-Platform:~/CS6233/assignments/lab2/dr3432hw2$ strace ./mycopy input.txt output.txt
execve("./mycopy", ["./mycopy", "input.txt", "output.txt"], 0x7ffd176c39f0 /* 48 vars */) = 0
brk(NULL)                               = 0x5ff6df001000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7a128d25d000
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=59099, ...}) = 0
mmap(NULL, 59099, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7a128d24e000
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\220\243\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
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 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
mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7a128d000000
mmap(0x7a128d028000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7a128d028000
mmap(0x7a128d1b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x7a128d1b0000
mmap(0x7a128d1ff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x7a128d1ff000
mmap(0x7a128d205000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7a128d205000
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7a128d24b000
arch_prctl(ARCH_SET_FS, 0x7a128d24b740) = 0
set_tid_address(0x7a128d24ba10)         = 3854
set_robust_list(0x7a128d24ba20, 24)     = 0
rseq(0x7a128d24c060, 0x20, 0, 0x53053053) = 0
mprotect(0x7a128d1ff000, 16384, PROT_READ) = 0
mprotect(0x5ff6decc6000, 4096, PROT_READ) = 0
mprotect(0x7a128d295000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7a128d24e000, 59099)           = 0
openat(AT_FDCWD, "input.txt", O_RDONLY) = 3
access("output.txt", F_OK)              = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "output.txt", O_WRONLY|O_CREAT, 0177610) = 4
read(3, "Hello world\nThis is lab 2\n\n\nHell"..., 255) = 34
write(4, "Hello world\nThis is lab 2\n\n\nHell"..., 34) = 34
read(3, "", 255)                        = 0
write(4, "\n\n", 2)                     = 2
write(4, "Diego Rosenberg\n", 16)       = 16
getuid()                                = 1000
write(4, "1000\0\0\0\0\0\0\0\0", 12)    = 12
write(4, "\n", 1)                       = 1
fchmod(4, 0644)                         = 0
close(3)                                = 0
close(4)                                = 0
exit_group(0)                           = ?
+++ exited with 0 +++
diegoros@diegoros-VMware-Virtual-Platform:~/CS6233/assignments/lab2/dr3432hw2$ 
```

Questions:

1) What are the system call names for getting the process' userID, opening a file, closing a file, reading a file and writing a file?
    - userID: getuid()
    - opening a file: openat()
    - closing a file: close()
    - reading a file: read()
    - writing a file: write()

2) How many system calls are (i.e. the count) involved with opening a file, closing a file, reading a file and writing a file? (count each individually. You may either use strace options to aid you in doing so, or you may use grep).

Each operation has 1 system call directly linked to it, i.e. for every openat, close, read and write operation you get 1 for every time you call the function. In this case since we opened 2 files, read the input file 2 times (since the file read once to completion and then once at EOF), wrote 5 times to the output file (could be reduced if we wanted to), and closed 2 files; we had 11 openat, read, write and close operations.

3) What was the value of the file descriptor of your read file? Should we expect it to change if you change the order of opening the input and output files?

The file descriptor of the read file ("input.txt") was 3 (it is convention to start file descriptor numbering at 3 for C since 0, 1, and 2 are linked with the standard input, output and error respectively). If we change the order in which we read the files then we will absolutely get different file descriptor order, as it is done sequentially. In this case if we swap the order then the fd for "input.txt" would have been 4.

4) What was the value of the file descriptor of your write file? Should we expect it to change if you change the order of opening the input and output files?

The file descriptor of the read file ("output.txt") was 4 (following the input file with fd=3). Once again, if we change the order in which we read the files then we will absolutely get different file descriptor order, as it is done sequentially. In this case if we swap the order then the fd for "output.txt" would have been 3.