BILKENT UNIVERSITY

CS 342 – OPERATING SYSTEMS

HOMEWORK #1 REPORT

Deniz Yüksel

21600880

1. Although I have a dualboot Ubuntu 18.04 with my Windows machine, I preferred to setup a virtual machine because in this homework, I have the feeling that we will experiment. So I do not want to lose my personalized Ubuntu. If I feel comfortable of myself at the end of this homework, I may consider using my own Ubuntu 18.04 for the second homework.

10 Linux commands:

mkdir “name”: This command creates a directory with the specifiec name in the quotation marks.

ls: This command lists all the user visible items in the current directory.

rm –r “directory_name”: This command removes all the items in a directory recursively, then removes the directory.

touch “file_name”: This command creates a file with the desired name.

cd .. : This command changes the current directory to one directory up.

history: This command shows in the CLI, what the user has executed in the past.

cd ~ : This command changes the current directory directly to home.

ls –la: This command shows all the files, including the hidden onces with the modify dates, their modifiability (read only, write only, read & write only etc.), and their locations (root, deniz, etc.)

mv “one_or_more_file_names” “path_name” : This command moves the specified files seperated with a space to a directory specified with a proper path name.

cp “one_or_more_file_names” “path_name” : This command copies and moves the specified files seperated with a space to a directory specified with a proper path name.

clear: This command clears the previous commands and processes in the CLI. If the user wishes to see the older commands again, he/she needs to use the mouse wheel to go up. In essence, this command just moves the old commands upwards which is invisible in the current terminal window.

man: This command gives information about a command or function in Linux.

2. The path of my executable kernel file, is in the root directory. It’s name is actually vmlinux but it is compressed to a file named vmlinuz. Vmlinuz gets uncompressed, loaded into the memory and executed at boot.

/vmlinuz

My version of Linux kernel is: 4.15.0-29-generic.

3. After I downloaded the linux 4.14.98 kernel source code, I unzipped the .xz file with the following command: tar xf linux-4.14.98.tar.xz.

The directories I see in this file are the following:

Arch, crypto, include, kernel, net, sound, block, Documentation, init, lib, tools, certs, drivers, ipc, samples, usr, firmware, scripts, virt, fs, mm, security.

There are other files that are not directories: README, MAINTAINERS, COPYING, Kbuild, Makefile, CREDITS, Kconfig.

4. The path that contains the system call table is the following:

~/Downloads/linux-4.14.98/arch/x86/entry/syscalls.

The file's name is syscall_64.tbl.

Syscall numbers:

5 – common, fstat, sys_newfstat

43 – common accept sys_accept

123 – common, setfsgid, sys_setfsgid

220 – common, semtimedop, sys_semtimedop

5. I recognized that when a proper command is executed with strace, exit_group() function is called with the parameter 0 in it, while a command with an error is called with exit_group(1).

When I execute strace with "clear" command, I get the following output:

```
execve("/usr/bin/clear", ["clear"], 0x7ffd04e08bf0 /* 59 vars */) = 0
brk(NULL)                       = 0x55fcb194d000
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=69886, ...}) = 0
mmap(NULL, 69886, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fa5c0017000
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/libtinfo.so.5", 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\311\0\0\0\0\0\0"..., 832) =
832

fstat(3, {st_mode=S_IFREG|0644, st_size=170784, ...}) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fa5c0015000

mmap(NULL, 2267936, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fa5bfbd8000

mprotect(0x7fa5bfbfd000, 2097152, PROT_NONE) = 0

mmap(0x7fa5bfdfd000, 20480, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7fa5bfdfd000

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) =
0x7fa5bf7e7000

mprotect(0x7fa5bf9ce000, 2097152, PROT_NONE) = 0

mmap(0x7fa5bfbce000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7fa5bfbce000

mmap(0x7fa5bfbd4000, 15072, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fa5bfbd4000

close(3)                        = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fa5c0012000

arch_prctl(ARCH_SET_FS, 0x7fa5c0012740) = 0

mprotect(0x7fa5bfbce000, 16384, PROT_READ) = 0

mprotect(0x7fa5bfdfd000, 16384, PROT_READ) = 0

mprotect(0x55fcafb9d000, 4096, PROT_READ) = 0

mprotect(0x7fa5c0029000, 4096, PROT_READ) = 0

munmap(0x7fa5c0017000, 69886)           = 0
```

```
ioctl(2, TCGETS, {B38400 opost isig icanon echo ...}) = 0
brk(NULL)                               = 0x55fcb194d000
brk(0x55fcb196e000)                     = 0x55fcb196e000
stat("/home/deniz/.terminfo", 0x55fcb194d430) = -1 ENOENT (No such file or directory)
stat("/etc/terminfo", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
stat("/lib/terminfo", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
stat("/usr/share/terminfo", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
access("/etc/terminfo/x/xterm-256color", R_OK) = -1 ENOENT (No such file or directory)
access("/lib/terminfo/x/xterm-256color", R_OK) = 0
openat(AT_FDCWD, "/lib/terminfo/x/xterm-256color", O_RDONLY) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=3525, ...}) = 0
read(3, "\32\1%\0&\0\17\0\235\1\2\6xterm-256color|xterm"..., 4096) = 3525
read(3, "", 4096)                       = 0
close(3)                                = 0
ioctl(2, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(2, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(2, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(2, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(2, TIOCGWINSZ, {ws_row=28, ws_col=79, ws_xpixel=0, ws_ypixel=0}) = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
write(1, "\33[3J\33[H\33[2J", 11)       = 11
exit_group(0)                           = ?
+++ exited with 0 +++
```

When I execute strace with “ls” command, I get the following output:

```
execve("/bin/ls", ["ls"], 0x7fff2fd5cb90 /* 59 vars */) = 0
brk(NULL)                               = 0x55a212942000
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=69886, ...}) = 0

mmap(NULL, 69886, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f2c504ae000

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/libselinux.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\20b\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=154832, ...}) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f2c504ac000

mmap(NULL, 2259152, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f2c50071000

mprotect(0x7f2c50096000, 2093056, PROT_NONE) = 0

mmap(0x7f2c50295000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x24000) = 0x7f2c50295000

mmap(0x7f2c50297000, 6352, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f2c50297000

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) =
0x7f2c4fc80000

mprotect(0x7f2c4fe67000, 2097152, PROT_NONE) = 0

mmap(0x7f2c50067000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f2c50067000

mmap(0x7f2c5006d000, 15072, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f2c5006d000

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/libpcre.so.3", 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 \25\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=464824, ...}) = 0

mmap(NULL, 2560264, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f2c4fa0e000

mprotect(0x7f2c4fa7e000, 2097152, PROT_NONE) = 0

mmap(0x7f2c4fc7e000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x70000) = 0x7f2c4fc7e000

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/libdl.so.2", 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\0P\16\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=14560, ...}) = 0

mmap(NULL, 2109712, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f2c4f80a000

mprotect(0x7f2c4f80d000, 2093056, PROT_NONE) = 0

mmap(0x7f2c4fa0c000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f2c4fa0c000

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, 2221184, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f2c4f5eb000

mprotect(0x7f2c4f605000, 2093056, PROT_NONE) = 0

mmap(0x7f2c4f804000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19000) = 0x7f2c4f804000

mmap(0x7f2c4f806000, 13440, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f2c4f806000

close(3)                        = 0
```

```
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f2c504aa000

arch_prctl(ARCH_SET_FS, 0x7f2c504ab040) = 0

mprotect(0x7f2c50067000, 16384, PROT_READ) = 0

mprotect(0x7f2c4f804000, 4096, PROT_READ) = 0

mprotect(0x7f2c4fa0c000, 4096, PROT_READ) = 0

mprotect(0x7f2c4fc7e000, 4096, PROT_READ) = 0

mprotect(0x7f2c50295000, 4096, PROT_READ) = 0

mprotect(0x55a211fc7000, 8192, PROT_READ) = 0

mprotect(0x7f2c504c0000, 4096, PROT_READ) = 0

munmap(0x7f2c504ae000, 69886)          = 0

set_tid_address(0x7f2c504ab310)        = 1635

set_robust_list(0x7f2c504ab320, 24)     = 0

rt_sigaction(SIGRTMIN, {sa_handler=0x7f2c4f5f0cb0, sa_mask=[],
sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7f2c4f5fd890}, NULL, 8) = 0

rt_sigaction(SIGRT_1, {sa_handler=0x7f2c4f5f0d50, sa_mask=[],
sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f2c4f5fd890}, 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("/sys/fs/selinux", 0x7ffd3c84dbf0) = -1 ENOENT (No such file or directory)

statfs("/selinux", 0x7ffd3c84dbf0)      = -1 ENOENT (No such file or directory)

brk(NULL)                              = 0x55a212942000

brk(0x55a212963000)                    = 0x55a212963000

openat(AT_FDCWD, "/proc/filesystems", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0444, st_size=0, ...}) = 0

read(3, "nodev\tsysfs\nnodev\trootfs\nnodev\tr"..., 1024) = 383

read(3, "", 1024)                      = 0

close(3)                               = 0

access("/etc/selinux/config", F_OK)     = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
```

```
fstat(3, {st_mode=S_IFREG|0644, st_size=11731760, ...}) = 0
mmap(NULL, 11731760, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f2c4eaba000
close(3)                        = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TIOCGWINSZ, {ws_row=28, ws_col=79, ws_xpixel=0, ws_ypixel=0}) = 0
openat(AT_FDCWD, ".", O_RDONLY|O_NONBLOCK|O_CLOEXEC|O_DIRECTORY) = 3
fstat(3, {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
getdents(3, /* 24 entries */, 32768)    = 784
getdents(3, /* 0 entries */, 32768)     = 0
close(3)                        = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
write(1, "deniz\t Documents  Music\tPictures"..., 44) = 44
write(1, "Desktop  Downloads  output.txt\tP"..., 47) = 47
close(1)                        = 0
close(2)                        = 0
exit_group(0)                   = ?
+++ exited with 0 +++
```

6. There are 3 time period reports as described in the assignment sheet: Real, user and sys. Real time is literally the real watch time that elapses from the start until the end of the call. User is the amount of time that is spent outside the kernel mode during the process, in other words in user mode. Sys is the time spent in kernel mode during the syscall. Therefore, user time plus the system time will be equivalent to the total time spent by the CPU.

clear time:

```
real	0m0,001s
user	0m0,000s
sys	0m0,001s
```

ls time:

```
real	0m0,001s
```

```
user    0m0,001s

sys     0m0,000s
```

mkdir time:

```
real    0m0,001s

user    0m0,001s

sys     0m0,000s
```

cp time:

```
real    0m0,002s

user    0m0,002s

sys     0m0,000s
```

7.	I will include my source code below. I have done 2 set of tests, in the first one I deleted all the files generated through “make” command and recompiled again, two is that I consecutively ran the executable output file over and over.

```
/*

        This is the C program experiment for 7th question.

        Deniz Yüksel

        21600880

        CS 342 - 3

*/


#include <stdio.h>

#include<sys/types.h>

#include<sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <stdlib.h>
```

```
#include <string.h>



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


	int fd;
	int sz;
	int capacity = 100;
	char *buffer = (char *) calloc(capacity, sizeof(char));
	struct timeval startOpen, stopOpen, startWrite, stopWrite, startClose, stopClose,
startOpen2, stopOpen2, startRead, stopRead;


	//Time to calculate open...
	printf("----- Open system call time calculation -----\n");
	gettimeofday(&startOpen, NULL);
	fd = open("myText.txt", O_CREAT | O_RDWR | O_EXCL, 0660);
	gettimeofday(&stopOpen, NULL);
	printf("Time elapsed for open: %ld %s", stopOpen.tv_usec - startOpen.tv_usec,
"microseconds\n");


	char text[] = "This text piece is written to the file now. It can take at most 100
characters.\n";
	char text2[] = "New text.\n";


	//Time to calculate write...
	printf("----- Write system call time calculation -----\n");
	gettimeofday(&startWrite, NULL);
	write(fd, text, strlen(text));
	gettimeofday(&stopWrite, NULL);
	printf("Time elapsed for write: %ld %s", stopWrite.tv_usec - startWrite.tv_usec,
"microseconds\n");
```

```
	//Time to calculate close...
	printf("----- Close system call time calculation -----\n");
	gettimeofday(&startClose, NULL);
	close(fd);
	gettimeofday(&stopClose, NULL);
	printf("Time elapsed for close: %ld %s", stopClose.tv_usec - startClose.tv_usec,
"microseconds\n");

	//Time to calculate open...
	printf("----- Measuring open system call again to read -----\n");
	gettimeofday(&startOpen2, NULL);
	fd = open("myText.txt", O_RDONLY);
	gettimeofday(&stopOpen2, NULL);
	printf("Time elapsed for open: %ld %s", stopOpen2.tv_usec - startOpen2.tv_usec,
"microseconds\n");

	//Time to calculate read...
	printf("----- Measuring read system call -----\n");
	gettimeofday(&startRead, NULL);
	sz = read(fd, buffer, capacity);
	gettimeofday(&stopRead, NULL);
	buffer[sz] = '\0';
	printf("Time elapsed for read: %ld %s", stopRead.tv_usec - startRead.tv_usec,
"microseconds\n");

	close(fd);
	printf("The text read is: %s\n", buffer);
	printf("Reached at the end!\n");
```

```
        return 0;
}
```

After the execution of the code for several times from scratch, meaning that I remove all the files at each iteration and compile again, these are the three most precise results:

```
----- Open system call time calculation -----
Time elapsed for open: 25 microseconds
----- Write system call time calculation -----
Time elapsed for write: 11 microseconds
----- Close system call time calculation -----
Time elapsed for close: 2 microseconds
----- Measuring open system call again to read -----
Time elapsed for open: 1 microseconds
----- Measuring read system call -----
Time elapsed for read: 1 microseconds
The text read is: This text piece is written to the file now. It can take at most 100 characters.

Reached at the end!
```

```
----- Open system call time calculation -----
Time elapsed for open: 22 microseconds
----- Write system call time calculation -----
Time elapsed for write: 10 microseconds
----- Close system call time calculation -----
Time elapsed for close: 1 microseconds
----- Measuring open system call again to read -----
Time elapsed for open: 2 microseconds
----- Measuring read system call -----
```

```
Time elapsed for read: 1 microseconds
The text read is: This text piece is written to the file now. It can take at most 100 characters.

Reached at the end!
```

```
----- Open system call time calculation -----
Time elapsed for open: 25 microseconds
----- Write system call time calculation -----
Time elapsed for write: 11 microseconds
----- Close system call time calculation -----
Time elapsed for close: 1 microseconds
----- Measuring open system call again to read -----
Time elapsed for open: 2 microseconds
----- Measuring read system call -----
Time elapsed for read: 1 microseconds
The text read is: This text piece is written to the file now. It can take at most 100 characters.

Reached at the end!
```

Note: If the output is ran consecutively without recompiling and deleting the text file, the time results are much less like this:

```
----- Open system call time calculation -----
Time elapsed for open: 3 microseconds
----- Write system call time calculation -----
Time elapsed for write: 1 microseconds
----- Close system call time calculation -----
Time elapsed for close: 1 microseconds
----- Measuring open system call again to read -----
```

```
Time elapsed for open: 2 microseconds

----- Measuring read system call -----

Time elapsed for read: 2 microseconds

The text read is: This text piece is written to the file now. It can take at most 100 characters.


Reached at the end!


              ----------------------------------------------------------------------------------------------------


----- Open system call time calculation -----

Time elapsed for open: 3 microseconds

----- Write system call time calculation -----

Time elapsed for write: 0 microseconds

----- Close system call time calculation -----

Time elapsed for close: 0 microseconds

----- Measuring open system call again to read -----

Time elapsed for open: 2 microseconds

----- Measuring read system call -----

Time elapsed for read: 1 microseconds

The text read is: This text piece is written to the file now. It can take at most 100 characters.


Reached at the end!


              ----------------------------------------------------------------------------------------------------


----- Open system call time calculation -----

Time elapsed for open: 3 microseconds

----- Write system call time calculation -----

Time elapsed for write: 1 microseconds
```

```
----- Close system call time calculation -----
Time elapsed for close: 1 microseconds
----- Measuring open system call again to read -----
Time elapsed for open: 2 microseconds
----- Measuring read system call -----
Time elapsed for read: 1 microseconds
The text read is: This text piece is written to the file now. It can take at most 100 characters.

Reached at the end!
```