Operating Systems

CS342 Hw1 Report

Fall 2018-2019

Aldo Tali

21500097

05.10.2018

The following is a report which answers the different sections of the first homework of Operating Systems Course Fall 2018.

Section-1

My choice for this exercise was the installation of a virtual machine box. The main reasons behind this came due to the fact that I do not currently my own laptop ad did not want to format any part of the already defined partitions of my machine and also because the laptop I am currently using has a dual boot (Windows and Ubuntu 17.0). Downgrading my already installed Ubuntu partition seemed illogical at these early steps and as such the Virtual Machine installation became my first choice for this task. In doing so I came to have only one particular problem which was the fact that during the installation the Oracle VM would not give me an option to install a 64bt version of the operating system (Ubuntu), it would offer only the 32-bit versions. At first I had to check if my current laptop is actually a 64-bit machine or not. It resulted to be so, and the actual problem was solved in the BIOS setup advanced configuration by enabling the "Acceptable Visualization Performance". After this I encountered no problems in my VM setup installation. Regarding the commands that were used on the system the following applies:

1.cd --> this command was used to change the directory. So suppose your current location is at ~/ and in your home directory you have a folder named "tester" then "cd tester" would make your new location be at ~/tester.

2.mv --> this command was used to move a file from a location to another. Suppose you are at location ~/loc1 containing test.txt and inside location ~/ two folders exist ~/loc2 and ~/loc1, then the command mv test.txt ./ ../loc2 would move the file "test.txt" to ~/loc2

3.pwd --> is a command which gives you the absolute path of the directory you are in. When you first open the terminal by default you are on the home directory so when I ran it I got /home/aldo.

4.ls --> is a command which lists all files in a directory.

5.touch --> is a command which allows the user to create a file in the current directory. This file can be anything for example you can type touch test.txt.

6.cp --> this is a command to copy files. For example cp test.txt /home/aldo copies the file test.txt in the current directory to the home directory.

7.locate --> this is a command used to find a file. For example locate test.txt would give /home/aldo provided that it exists in that location.

8.cat --> is a command generally used for displaying the contents of a file. For example cat test.txt shows the contents of text.txt in the console.

9.df --> is a command that shows the available disk space in every partition. All you need to do is press df in the terminal.

10.rm --> is a command used to remove files from the Linux systems. Example rm test.txt.

Section-2

The location of the Linux Kernel executable is /boot.

The version of it is : 4.15.0-29-generic

Section-3

The directories found here are :

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

Section-4

Sample output for ls:

```
execve("/bin/ls", ["ls"], [/* 90 vars */]) = 0
brk(NULL)                           = 0x561809b93000
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=99041, ...}) = 0
mmap(NULL, 99041, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fe841063000
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) =
0x7fe841080000
mmap(NULL, 2259152, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fe840c33000
mprotect(0x7fe840c58000, 2093056, PROT_NONE) = 0
mmap(0x7fe840e57000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x24000) = 0x7fe840e57000
mmap(0x7fe840e59000, 6352, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7fe840e59000
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\340\22\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1960656, ...}) = 0
mmap(NULL, 4061792, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fe840853000
mprotect(0x7fe840a29000, 2097152, PROT_NONE) = 0
mmap(0x7fe840c29000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1d6000) = 0x7fe840c29000
mmap(0x7fe840c2f000, 14944, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7fe840c2f000
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) =
0x7fe8405db000

mprotect(0x7fe84064b000, 2097152, PROT_NONE) = 0

mmap(0x7fe84084b000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x70000) = 0x7fe84084b000

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\0\220\16\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=14632, ...}) = 0

mmap(NULL, 2109712, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fe8403d3000

mprotect(0x7fe8403d6000, 2093056, PROT_NONE) = 0

mmap(0x7fe8405d5000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2000) = 0x7fe8405d5000

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\0\360a\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0755, st_size=144776, ...}) = 0

mmap(NULL, 2221160, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fe8401b3000

mprotect(0x7fe8401cd000, 2093056, PROT_NONE) = 0

mmap(0x7fe8403cc000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x19000) = 0x7fe8403cc000

mmap(0x7fe8403ce000, 13416, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7fe8403ce000

close(3)                         = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fe84107e000

arch_prctl(ARCH_SET_FS, 0x7fe84107f040) = 0

mprotect(0x7fe840c29000, 16384, PROT_READ) = 0

mprotect(0x7fe8403cc000, 4096, PROT_READ) = 0

mprotect(0x7fe8405d5000, 4096, PROT_READ) = 0

mprotect(0x7fe84084b000, 4096, PROT_READ) = 0
mprotect(0x7fe840e57000, 4096, PROT_READ) = 0
mprotect(0x561808b44000, 4096, PROT_READ) = 0
mprotect(0x7fe841082000, 4096, PROT_READ) = 0
munmap(0x7fe841063000, 99041)           = 0
set_tid_address(0x7fe84107f310)         = 8235
set_robust_list(0x7fe84107f320, 24)     = 0
rt_sigaction(SIGRTMIN, {sa_handler=0x7fe8401b8c70, sa_mask=[], sa_flags=SA_RESTORER|
SA_SIGINFO, sa_restorer=0x7fe8401c6150}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {sa_handler=0x7fe8401b8d00, sa_mask=[], sa_flags=SA_RESTORER|
SA_RESTART|SA_SIGINFO, sa_restorer=0x7fe8401c6150}, 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", 0x7ffd0b08f000) = -1 ENOENT (No such file or directory)
statfs("/selinux", 0x7ffd0b08f000)      = -1 ENOENT (No such file or directory)
brk(NULL)                       = 0x561809b93000
brk(0x561809bb4000)                     = 0x561809bb4000
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)
open("/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=4446656, ...}) = 0
mmap(NULL, 4446656, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fe83fd73000
close(3)                        = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TIOCGWINSZ, {ws_row=38, ws_col=150, ws_xpixel=0, ws_ypixel=0}) = 0
open(".", O_RDONLY|O_NONBLOCK|O_DIRECTORY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
getdents(3, /* 9 entries */, 32768)     = 272
getdents(3, /* 0 entries */, 32768)     = 0
close(3)                        = 0
```

```
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
write(1, "answers.txt  cost  cost.c  ex4.t"..., 67answers.txt  cost  cost.c  ex4.txt  ex5.c  Homework1.pdf  makefile
) = 67
close(1)                                = 0
close(2)                                = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```

Sample output for cp:

```
execve("/bin/cp", ["cp"], [/* 90 vars */]) = 0
brk(NULL)                               = 0x55f2ba039000
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=99041, ...}) = 0
mmap(NULL, 99041, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f250e48b000
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) = 0x7f250e4ae000
mmap(NULL, 2259152, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f250e05b000
mprotect(0x7f250e080000, 2093056, PROT_NONE) = 0
mmap(0x7f250e27f000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x24000) = 0x7f250e27f000
mmap(0x7f250e281000, 6352, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f250e281000
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/libacl.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\340\33\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=31232, ...}) = 0
mmap(NULL, 2126336, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f250de53000
mprotect(0x7f250de5a000, 2093056, PROT_NONE) = 0
mmap(0x7f250e059000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x6000) = 0x7f250e059000
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/libattr.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\260\20\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=18680, ...}) = 0
mmap(NULL, 2113752, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f250dc4b000
mprotect(0x7f250dc4f000, 2093056, PROT_NONE) = 0
mmap(0x7f250de4e000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x3000) = 0x7f250de4e000
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\340\22\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1960656, ...}) = 0
mmap(NULL, 4061792, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f250d86b000
mprotect(0x7f250da41000, 2097152, PROT_NONE) = 0
mmap(0x7f250dc41000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1d6000) = 0x7f250dc41000
mmap(0x7f250dc47000, 14944, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7f250dc47000
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) =
0x7f250d5f3000
mprotect(0x7f250d663000, 2097152, PROT_NONE) = 0
```

```
mmap(0x7f250d863000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x70000) = 0x7f250d863000

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\0\220\16\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0644, st_size=14632, ...}) = 0

mmap(NULL, 2109712, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f250d3eb000

mprotect(0x7f250d3ee000, 2093056, PROT_NONE) = 0

mmap(0x7f250d5ed000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2000) = 0x7f250d5ed000

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\0\360a\0\0\0\0\0\0"..., 832) = 832

fstat(3, {st_mode=S_IFREG|0755, st_size=144776, ...}) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f250e4a8000

mmap(NULL, 2221160, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f250d1cb000

mprotect(0x7f250d1e5000, 2093056, PROT_NONE) = 0

mmap(0x7f250d3e4000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x19000) = 0x7f250d3e4000

mmap(0x7f250d3e6000, 13416, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7f250d3e6000

close(3)                    = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0x7f250e4a5000

arch_prctl(ARCH_SET_FS, 0x7f250e4a5800) = 0

mprotect(0x7f250dc41000, 16384, PROT_READ) = 0

mprotect(0x7f250d3e4000, 4096, PROT_READ) = 0

mprotect(0x7f250d5ed000, 4096, PROT_READ) = 0

mprotect(0x7f250d863000, 4096, PROT_READ) = 0

mprotect(0x7f250de4e000, 4096, PROT_READ) = 0

mprotect(0x7f250e059000, 4096, PROT_READ) = 0

mprotect(0x7f250e27f000, 4096, PROT_READ) = 0
mprotect(0x55f2b92d8000, 4096, PROT_READ) = 0
mprotect(0x7f250e4aa000, 4096, PROT_READ) = 0
munmap(0x7f250e48b000, 99041)           = 0
set_tid_address(0x7f250e4a5ad0)         = 8259
set_robust_list(0x7f250e4a5ae0, 24)     = 0
rt_sigaction(SIGRTMIN, {sa_handler=0x7f250d1d0c70, sa_mask=[], sa_flags=SA_RESTORER|
SA_SIGINFO, sa_restorer=0x7f250d1de150}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {sa_handler=0x7f250d1d0d00, sa_mask=[], sa_flags=SA_RESTORER|
SA_RESTART|SA_SIGINFO, sa_restorer=0x7f250d1de150}, 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", 0x7ffccf782e70) = -1 ENOENT (No such file or directory)
statfs("/selinux", 0x7ffccf782e70)      = -1 ENOENT (No such file or directory)
brk(NULL)                       = 0x55f2ba039000
brk(0x55f2ba05a000)                     = 0x55f2ba05a000
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)
open("/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=4446656, ...}) = 0
mmap(NULL, 4446656, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f250cd8b000
close(3)                        = 0
geteuid()                       = 1000
open("/usr/share/locale/locale.alias", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=2995, ...}) = 0
read(3, "# Locale name alias data base.\n#"..., 4096) = 2995
read(3, "", 4096)               = 0
close(3)                        = 0
open("/usr/share/locale/en_US/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (No
such file or directory)
open("/usr/share/locale/en/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1 ENOENT (No such
file or directory)
```

```
open("/usr/share/locale-langpack/en_US/LC_MESSAGES/coreutils.mo", O_RDONLY) = -1
ENOENT (No such file or directory)
open("/usr/share/locale-langpack/en/LC_MESSAGES/coreutils.mo", O_RDONLY) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=578, ...}) = 0
mmap(NULL, 578, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f250e4a3000
close(3)                        = 0
write(2, "cp: ", 4cp: )                   = 4
write(2, "missing file operand", 20missing file operand)         = 20
write(2, "\n", 1
)                       = 1
write(2, "Try 'cp --help' for more informa"..., 38Try 'cp --help' for more information.
) = 38
lseek(0, 0, SEEK_CUR)                   = -1 ESPIPE (Illegal seek)
close(0)                        = 0
close(1)                        = 0
close(2)                        = 0
exit_group(1)                   = ?
+++ exited with 1 +++
```

Section-5

The following represents a table of time keeping in microseconds for each of the tested system calls.

write()

Parameters	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Mean	S.Deviation
100 bytes	7	5	7	8	7	6	6.667	0.942
1000 bytes	25	16	24	25	32	21	23.83	4.811

10000 bytes	24	29	20	28	50	22	28.83	9.97
100000 bytes	64	47	61	59	59	53	57.17	5.611

read()

Parameters	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Mean	S.Deviation
100 bytes	9	11	11	9	5	10	9.167	2.304
1000 bytes	7	9	9	13	16	14	11.33	3.197
10000 bytes	23	24	25	26	47	34	29.83	8.474
100000 bytes	106	128	106	120	83	141	114	18.48

open()

Parameters	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Mean	S.Deviation
Read only	35	27	33	40	38	29	33.67	4.607
Write only	27	27	30	36	30	22	28.67	4.230
Read and Write	28	28	29	35	29	21	28.33	4.068

Read Write with Data Erase	29	31	30	31	37	22	30	4.397

remove()

Parameters	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Mean	S.Deviation
100 bytes	14	20	95	64	64	63	53.33	28.034
1000 bytes	27	36	91	31	43	42	45	21.33
10000 bytes	47	56	36	18	24	25	34.33	13.499
100000 bytes	90	114	82	120	93	109	101.3	13.78

mkdir()

Parameters	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Mean	S.Deviation
Default	76	67	75	62	78	61	69.83	6.81705

getpid()

Parameters	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Mean	S.Deviation
Default	0	1	1	0	0	1	0.5	0.5

This is the source code:

```
/*

 *

 * #####################################################

 * Date: 05/10/2018

 * @Author: Aldo Tali 21500097

 * Description: Solution to part 5 of the first

 *  homework in Cs342 Operating Systems Fall 2018-2019.

 * #####################################################

 *

 * */




#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <sys/time.h>

#include <stdlib.h>

#include <string.h>
```

```
#include <fcntl.h>
#include <sys/stat.h>

//The following are the function routine signatures that simulate
the function calls with different parameters
//
//simulates the file creation to write files with different sizes
void createFileWithSizeB(int size, FILE *file);
//simulates the file reading of files with different sizes
void checkReadFileSizeB(int size);
//simulates the opening time of files with different sizes
void checkOpenCommand();
//simulates the delete time of files with different sizes
void checkRemoveCommand();
//simualtes the time of make directory command
void checkMkdirCommand();
void displayPid();

//writes file to a given size
void createFileWithSizeB(int size, FILE *file)
{
    // int pid = getpid();
    //printf("The PID in Routine createFile with fileSize %d bytes
is %d\n",size,pid);

    //keep the current time
    struct timeval  tv;
    gettimeofday(&tv,NULL);

    //write enough to fill the file up to a given file size
    fseek(file,size-1,SEEK_SET);
    fputc('\0', file);
    fclose(file);

    // get the finish time
    struct timeval  tv2;
    gettimeofday(&tv2,NULL);

    //get numerical values of the times and print them out
    long long startTime = tv.tv_sec*10000000 + tv.tv_usec;
    long long endTime = tv2.tv_sec*10000000 + tv2.tv_usec;

    printf("The Time needed to create %d bytes, was %lld microseconds.\n",size, (endTime-startTime));

}

//reads file of a given size of bytes
//CRUCIAL: Be careful it makes use of the files created on the previous function
void checkReadFileSizeB(int size)
{
    //int pid = getpid();
    //printf("The PID in Routine checkRead with file size %d bytes
is %d\n",size,pid);

    //use this for the file read
    char *buffer =  (char *) calloc(size, sizeof(char));
    char str[50];

    sprintf(str, "%d", size);

    //record the beggining time
    struct timeval  tv;
    gettimeofday(&tv,NULL);

    int fp = open(strcat(str, "bytes"), O_RDONLY);
    int bytes = read(fp,buffer,size);

    close(fp);
    struct timeval  tv2;
    gettimeofday(&tv2,NULL);

    //display tun time
    long long startTime = tv.tv_sec*10000000 + tv.tv_usec;
    long long endTime = tv2.tv_sec*10000000 + tv2.tv_usec;
    printf("The Time needed to read %d bytes, was %lld microseconds.\n",bytes, (endTime-startTime));

}

//simulates the open command of linux
void checkOpenCommand()
{
    //record begining time
    struct timeval  tv,tv2;
    gettimeofday(&tv,NULL);
    long long startTime = tv.tv_sec*10000000 + tv.tv_usec;

    //opem the file and print the time of read
    printf("Opening File Read Only Mode\n");
    int description = open("checkOpenReadOnly",O_RDONLY | O_CREAT);
    gettimeofday(&tv2,NULL);
    long long endTime = tv2.tv_sec*10000000 + tv2.tv_usec;
    printf("The Time elapsed to open the files was : %lld microseconds\n", endTime - startTime);



    //repeat same process for different modes
    gettimeofday(&tv,NULL);
    startTime = tv.tv_sec*10000000 + tv.tv_usec;
    printf("Opening File Write Only Mode\n");
    description = open("checkOpenWriteOnly",O_WRONLY | O_CREAT);
    gettimeofday(&tv2,NULL);
    endTime = tv2.tv_sec*10000000 + tv2.tv_usec;
    printf("The Time elapsed to open the files was : %lld microseconds\n", endTime - startTime);

    gettimeofday(&tv,NULL);
    startTime = tv.tv_sec*10000000 + tv.tv_usec;
    printf("Opening File Read and Write  Mode\n");
    description = open("checkOpenReadWrite", O_RDWR | O_CREAT);
    gettimeofday(&tv2,NULL);
    endTime = tv2.tv_sec*10000000 + tv2.tv_usec;
    printf("The Time elapsed to open the files was : %lld microseconds\n", endTime - startTime);

    gettimeofday(&tv,NULL);
    startTime = tv.tv_sec*10000000 + tv.tv_usec;
    printf("Opening File Read and Write  Mode by erasing all inital info \n");
    description = open("checkOpenReadWriteWithTruncate",O_RDWR | O_TRUNC | O_CREAT);

    gettimeofday(&tv2,NULL);
    endTime = tv2.tv_sec*10000000 + tv2.tv_usec;
    printf("The Time elapsed to open the files was : %lld microseconds\n", endTime - startTime);

}

//simulates the timings for the remove/delete comand
void checkRemoveCommand()
{
    //record beginign time
    struct timeval  tv,tv2;
    gettimeofday(&tv,NULL);
    long long startTime = tv.tv_sec*10000000 + tv.tv_usec;

    //remove the files created before on the read modes
    printf("Deleting File Read Only Mode\n");
    remove("checkOpenReadOnly");
    printf("Deleting File Write Only Mode\n");
    remove("checkOpenWriteOnly");
    printf("Deleting File Read and Write  Mode\n");
    remove("checkOpenReadWrite");
    printf("Deleteing File Read and Write  Mode by erasing all inital info \n");
    remove("checkOpenReadWriteWithTruncate");

    gettimeofday(&tv2,NULL);
    long long endTime = tv2.tv_sec*10000000 + tv2.tv_usec;
    printf("The Time elapsed to remove the files was : %lld microseconds\n", endTime - startTime);

    //check timings for removing each of the 100, 1000 or 10000 bytes files
    gettimeofday(&tv,NULL);
    startTime = tv.tv_sec*10000000 + tv.tv_usec;
    printf("Deleting 100000 bytes file \n");
    remove("100000bytes");
    gettimeofday(&tv2,NULL);
    endTime = tv2.tv_sec*10000000 + tv2.tv_usec;
    printf("The Time elapsed to remove the files was : %lld microseconds \n", endTime - startTime);

    gettimeofday(&tv,NULL);
    startTime = tv.tv_sec*10000000 + tv.tv_usec;
    printf("Deleting 10000 bytes file \n");
    remove("10000bytes");
    gettimeofday(&tv2,NULL);
    endTime = tv2.tv_sec*10000000 + tv2.tv_usec;
    printf("The Time elapsed to remove the files was : %lld microseconds \n", endTime - startTime);

    gettimeofday(&tv,NULL);
    startTime = tv.tv_sec*10000000 + tv.tv_usec;
    printf("Deleting 1000 bytes file \n");
    remove("1000bytes");
    gettimeofday(&tv2,NULL);
    endTime = tv2.tv_sec*10000000 + tv2.tv_usec;
    printf("The Time elapsed to remove the files was : %lld microseconds\n", endTime - startTime);

    gettimeofday(&tv,NULL);
    startTime = tv.tv_sec*10000000 + tv.tv_usec;
    printf("Deleting 100 bytes file \n");
    remove("100bytes");
    gettimeofday(&tv2,NULL);
    endTime = tv2.tv_sec*10000000 + tv2.tv_usec;
    printf("The Time elapsed to remove the files was : %lld microseconds\n", endTime - startTime);

}

//simulates the time requirements for the make directory command
void checkMkdirCommand()
{
    //keep the begin time
    struct timeval  tv,tv2;
    gettimeofday(&tv,NULL);
    long long startTime = tv.tv_sec*10000000 + tv.tv_usec;
    //create directory
    printf("Creating Directory \n");
    int description = mkdir("ex5", 0777);
    //record end time
    gettimeofday(&tv2,NULL);
    long long endTime = tv2.tv_sec*10000000 + tv2.tv_usec;

    printf("The Time elapsed to create the directory was : %lld microseconds \n", endTime - startTime);
    rmdir("ex5");

}

void displayPid()
{
    int pid = getpid();
    printf("The PID in Routine is %d\n",pid);

    //keep the current time
    struct timeval  tv;
    gettimeofday(&tv,NULL);

    // get the finish time
    struct timeval  tv2;
    gettimeofday(&tv2,NULL);
    //get numerical values of the times and print them out
    long long startTime = tv.tv_sec*10000000 + tv.tv_usec;
    long long endTime = tv2.tv_sec*10000000 + tv2.tv_usec;

    printf("The Time needed to get pid, was %lld microseconds.\n",
(endTime-startTime));

}

//run the tester
int main()
{
    //4 files needed to serve for the different parameters
    FILE *file = fopen("100000bytes", "w");
    FILE *file2 = fopen("10000bytes", "w");
    FILE *file3 = fopen("1000bytes", "w");
    FILE *file4 = fopen("100bytes", "w");

    // create the files with different sizes and record their tim-
ings accordingly
    printf("--------------------------------------------------------
----------------------------------\n");
    createFileWithSizeB(100000, file);
    createFileWithSizeB(10000, file2);
    createFileWithSizeB(1000, file3);
    createFileWithSizeB(100, file4);
    printf("--------------------------------------------------------
----------------------------------\n");

    // read the files with different sizes and record their timings
accordingly
    printf("\n\n----------------------------------------------------
--------------------------------------\n");
    checkReadFileSizeB(100000);
    checkReadFileSizeB(10000);
    checkReadFileSizeB(1000);
    checkReadFileSizeB(100);
    printf("--------------------------------------------------------
----------------------------------\n");



    // open the files with different parameters and record their
timings accordingly
    printf("\n\n----------------------------------------------------
--------------------------------------\n");
    checkOpenCommand();
    printf("--------------------------------------------------------
----------------------------------\n");

    // remove the files with different sizes and record their tim-
ings accordingly
```

```
    printf("\n\n----------------------------------------------------
--------------------------------------\n");
    checkRemoveCommand();
    printf("--------------------------------------------------------
----------------------------------\n");

    // check the make directory timing accordingly
    printf("\n\n----------------------------------------------------
--------------------------------------\n");
    checkMkdirCommand();
    printf("--------------------------------------------------------
----------------------------------\n");



    printf("\n\n----------------------------------------------------
--------------------------------------\n");
    displayPid();
    printf("--------------------------------------------------------
----------------------------------\n");





    return 0;
}
```