1. Create input.txt

```
mycroft@mycroft-VMware-Virtual-Platform:~$ touch input.txt
mycroft@mycroft-VMware-Virtual-Platform:~$ echo "Hello world"        >  input.txt
echo "This is lab 2"      >> input.txt
echo "Operating Systems!" >> input.txt
```

2. Create mycat.c

```
mycroft@mycroft-VMware-Virtual-Platform:~$ cat > mycat.c <<'EOF'
#define _POSIX_C_SOURCE 200809L
#include <unistd.h>     // getpid, read, write, close, sleep
#include <fcntl.h>      // open, O_RDONLY
#include <sys/types.h>
#include <sys/stat.h>
#include <errno.h>
#include <time.h>       // time
#include <stdlib.h>     // srand, rand, exit
#include <string.h>     // strerror
#include <stdio.h>      // dprintf, snprintf
```

3. compile and run

```
mycroft@mycroft-VMware-Virtual-Platform:~$ gcc -Wall -Wextra -O2 -std=c11 -o mycat mycat.c
mycroft@mycroft-VMware-Virtual-Platform:~$ ./mycat input.txt
4972
Hello world
This is lab 2
Operating Systems!
```

4. Change input content and run

```
mycroft@mycroft-VMware-Virtual-Platform:~$ ./mycat input.txt
5013
Hello world
This is lab 2 from Hongdao Meng
Operating Systems!
```

1) What are the system call names for getting the process ID, opening a file, closing a file, reading a file, printing to the console and sleeping?

```
mycroft@mycroft-VMware-Virtual-Platform:~$ strace ./mycat input.txt
execve("./mycat", ["./mycat", "input.txt"], 0x7ffebeb686c8 /* 49 vars */) = 0
brk(NULL)                               = 0x58f67cf98000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7174e05bc000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (没有那个文件或目录)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=75175, ...}) = 0
mmap(NULL, 75175, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7174e05a9000
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"..., 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) = 0x7174e0200000
mmap(0x7174e0228000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7174e0228000
mmap(0x7174e03b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x7174e03b0000
mmap(0x7174e03ff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x7174e03ff000
mmap(0x7174e0405000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7174e0405000
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7174e05a6000
arch_prctl(ARCH_SET_FS, 0x7174e05a6740) = 0
set_tid_address(0x7174e05a6a10)         = 5068
set_robust_list(0x7174e05a6a20, 24)     = 0
rseq(0x7174e05a7060, 0x20, 0, 0x53053053) = 0
mprotect(0x7174e03ff000, 16384, PROT_READ) = 0
mprotect(0x58f644e19000, 4096, PROT_READ) = 0
mprotect(0x7174e05fa000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7174e05a9000, 75175)           = 0
getpid()                                = 5068
write(1, "5068\n", 55068
)                       = 5
clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=2, tv_nsec=0}, 0x7fff1b35ddb0) = 0
openat(AT_FDCWD, "input.txt", O_RDONLY) = 3
read(3, "Hello world\nThis is lab 2 from H"..., 4096) = 63
write(1, "Hello world\nThis is lab 2 from H"..., 63Hello world
This is lab 2 from Hongdao Meng
Operating Systems!
) = 63
read(3, "", 4096)                       = 0
close(3)                                = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```

Getting the process ID → getpid
Opening a file → openat
Closing a file → close
Reading a file → read
Printing to the console → write
Sleeping → nanosleep

2) What are the number of system calls for opening, closing and reading the file(s) (i.e. how many times each was called).

```
mycroft@mycroft-VMware-Virtual-Platform:~$ strace -c ./mycat input.txt
5133
Hello world
This is lab 2 from Hongdao Meng
Operating Systems!
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 49.67    0.000149         149         1           execve
 10.33    0.000031          15         2           write
 10.33    0.000031           3         8           mmap
  6.67    0.000020           6         3           openat
  6.33    0.000019          19         1           clock_nanosleep
  4.00    0.000012           4         3           mprotect
  3.00    0.000009           3         3           read
  2.67    0.000008           8         1           munmap
  2.00    0.000006           2         3           close
  1.00    0.000003           1         2           fstat
  1.00    0.000003           3         1         1 access
  0.67    0.000002           1         2           pread64
  0.33    0.000001           1         1           brk
  0.33    0.000001           1         1           getpid
  0.33    0.000001           1         1           arch_prctl
  0.33    0.000001           1         1           set_tid_address
  0.33    0.000001           1         1           set_robust_list
  0.33    0.000001           1         1           prlimit64
  0.33    0.000001           1         1           rseq
------ ----------- ----------- --------- --------- ----------------
100.00    0.000300           8        37         1 total
```

openat → 3
close → 3
read → 3

3) What are the number of system calls for printing to the screen? (count each individually. You may either use strace options to aid you in doing so, or you may use grep).

Printing to the screen use ‘write’
write→ 2

4) What was the value of the file descriptor of your read file (please review the lecture slides before asking what this means)?

```
)                       = 5
clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=2, tv_nsec=0}, 0x7fff1b35ddb0) = 0
openat(AT_FDCWD, "input.txt", O_RDONLY) = 3
read(3, "Hello world\nThis is lab 2 from H"..., 4096) = 63
write(1, "Hello world\nThis is lab 2 from H"..., 63Hello world
```

The value of the file descriptor for the read file (input.txt) was 3.