CPSC 457 – FALL 2022

Assignment: 1

Name: Farica Mago

UCID: 30111924

Q1 - Written question (5 marks)

a) Use the time utility to time palindrome.py and slow-pali.cpp on files t4.txt and t3.txt. Copy/paste the output of time from the terminal window into your report.

```
farica.mago@linux12-ec:~/palindrome$ time python3 palindrome.py < t4.txt
Longest palindrome: redder

real    0m0.233s
user    0m0.216s
sys     0m0.011s
farica.mago@linux12-ec:~/palindrome$ time python3 palindrome.py < t3.txt
Longest palindrome: ___o.0.o___

real    0m0.022s
user    0m0.016s
sys     0m0.004s
farica.mago@linux12-ec:~/palindrome$
```

```
farica.mago@linux12-ec:~/palindrome$ make
g++ -O2 -Wall slow-pali.cpp -o slow-pali
g++ -O2 -Wall fast-pali.cpp -o fast-pali
farica.mago@linux12-ec:~/palindrome$ time ./slow-pali < t4.txt
Longest palindrome: redder

real    0m3.740s
user    0m1.693s
sys     0m2.041s
farica.mago@linux12-ec:~/palindrome$ time ./slow-pali < t3.txt
Longest palindrome: ___o.0.o___

real    0m0.008s
user    0m0.005s
sys     0m0.003s
farica.mago@linux12-ec:~/palindrome$
```

b) How much time did the C++ and python programs spend in kernel vs user mode?

For python programs:

- For t4.txt, the time spent in kernel mode is: 0.011 seconds and the time spent in user mode: 0.216 seconds.
- For t3.txt, the time spent in kernel mode is: 0.004 seconds and the time spent in user mode: 0.016 seconds.

For C++ programs:

- For t4.txt, the time spent in kernel mode is: 2.041 seconds and the time spent in user mode: 1.693 seconds.
- For t3.txt, the time spent in kernel mode is: 0.003 seconds and the time spent in user mode: 0.005 seconds.

c) Run 'strace -c' on palindrome.py and slow-pali.cpp on t4.txt and t3.txt. Copy/paste the output from the terminal window into your report.

```
farica.mago@linux12-ec:~/palindrome$ strace -c ./palindrome.py < t4.txt
Longest palindrome: redder
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 26.90    0.000411           1       252        38 newfstatat
 14.92    0.000228           2        84        18 openat
 10.54    0.000161          32         5         3 execve
  9.62    0.000147           3        46           mmap
  9.55    0.000146           9        16           getdents64
  7.13    0.000109           0       784           read
  6.48    0.000099           1        69           close
  2.49    0.000038           0        70         2 lseek
  2.36    0.000036           4         8           mprotect
  1.83    0.000028           0        45        40 ioctl
  1.57    0.000024           0        26           brk
  1.37    0.000021           4         5           munmap
  0.92    0.000014           1         9           pread64
  0.59    0.000009           3         3           getrandom
  0.46    0.000007           1         4         2 arch_prctl
  0.39    0.000006           3         2           set_robust_list
  0.33    0.000005           1         4         3 readlink
  0.33    0.000005           2         2           set_tid_address
  0.33    0.000005           2         2           prlimit64
  0.26    0.000004           2         2         2 access
  0.26    0.000004           4         1           sysinfo
  0.26    0.000004           2         2           futex
  0.26    0.000004           2         2           rseq
  0.20    0.000003           1         2           getcwd
  0.20    0.000003           3         1           geteuid
  0.20    0.000003           3         1           getegid
  0.13    0.000002           2         1           getuid
  0.13    0.000002           2         1           getgid
  0.00    0.000000           0         1           write
  0.00    0.000000           0        66           rt_sigaction
  0.00    0.000000           0         3           dup
  0.00    0.000000           0         1           fcntl
------ ----------- ----------- --------- --------- ----------------
100.00    0.001528           1      1520       108 total
```

```
farica.mago@linux12-ec:~/palindrome$ strace -c ./palindrome.py < t3.txt
Longest palindrome: ___o.O.o___
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 16.89    0.000211           0       252        38 newfstatat
 16.57    0.000207           2        84        18 openat
 12.73    0.000159          31         5         3 execve
 12.01    0.000150           3        46           mmap
  7.69    0.000096           1        69           close
  6.57    0.000082           1        80           read
  6.41    0.000080           5        16           getdents64
  2.88    0.000036           0        70         2 lseek
  2.88    0.000036           4         8           mprotect
  2.80    0.000035           0        45        40 ioctl
  2.32    0.000029           0        66           rt_sigaction
  2.24    0.000028           2        12           brk
  1.60    0.000020           4         5           munmap
  1.04    0.000013           1         9           pread64
  0.96    0.000012           4         3           getrandom
  0.80    0.000010           3         3           dup
  0.48    0.000006           3         2           set_tid_address
  0.48    0.000006           3         2           rseq
  0.40    0.000005           2         2         2 access
  0.40    0.000005           2         2           futex
  0.40    0.000005           2         2           prlimit64
  0.32    0.000004           2         2           getcwd
  0.32    0.000004           1         4         3 readlink
  0.32    0.000004           1         4         2 arch_prctl
  0.32    0.000004           2         2           set_robust_list
  0.16    0.000002           2         1           fcntl
  0.00    0.000000           0         1           write
  0.00    0.000000           0         1           sysinfo
  0.00    0.000000           0         1           getuid
  0.00    0.000000           0         1           getgid
  0.00    0.000000           0         1           geteuid
  0.00    0.000000           0         1           getegid
------ ----------- ----------- --------- --------- ----------------
100.00    0.001249           1       802       108 total
```

```
farica.mago@linux12-ec:~/palindrome$ $make
farica.mago@linux12-ec:~/palindrome$ strace -c ./slow-pali < t4.txt
Longest palindrome: redder
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
100.00   12.588532           2   5767198           read
  0.00    0.000006           6         1           write
  0.00    0.000005           0         6           newfstatat
  0.00    0.000000           0         5           close
  0.00    0.000000           0        23           mmap
  0.00    0.000000           0         7           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         5           pread64
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         2         1 arch_prctl
  0.00    0.000000           0         1           set_tid_address
  0.00    0.000000           0         5           openat
  0.00    0.000000           0         1           set_robust_list
  0.00    0.000000           0         1           prlimit64
  0.00    0.000000           0         1           getrandom
  0.00    0.000000           0         1           rseq
------ ----------- ----------- --------- --------- ----------------
100.00   12.588543           2   5767263         2 total
farica.mago@linux12-ec:~/palindrome$
```

```
farica.mago@linux12-ec:~/palindrome$ strace -c ./slow-pali < t3.txt
Longest palindrome: ___o.O.o___
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 35.59    0.000079           3        23           mmap
 27.03    0.000060           1        43           read
  8.56    0.000019           2         7           mprotect
  6.31    0.000014           2         5           openat
  4.50    0.000010           2         5           pread64
  4.05    0.000009           1         6           newfstatat
  3.60    0.000008           1         5           close
  2.25    0.000005           5         1           munmap
  1.35    0.000003           3         1           write
  1.35    0.000003           1         3           brk
  1.35    0.000003           1         2         1 arch_prctl
  0.90    0.000002           2         1           set_tid_address
  0.90    0.000002           2         1           set_robust_list
  0.90    0.000002           2         1           getrandom
  0.90    0.000002           2         1           rseq
  0.45    0.000001           1         1           prlimit64
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
------ ----------- ----------- --------- --------- ----------------
100.00    0.000222           2       108         2 total
farica.mago@linux12-ec:~/palindrome$
```

d) When compared to the C++ code, why is the python program faster on some inputs, and slower on others? Try to justify your answers using the results you obtained above.

For t4.txt, python program is faster than C++ program because, in python program the read() function is called 784 times whereas the read() function in C++ program is called 5767198 times.

For t3.txt, python programs is slower than C++ program because, in python program, the read() function is called 80 times whereas, the read() function in C++ program is called 43 times.

Q1 - Written question (5 marks)

a) **Run your fast-pali.cpp on t3.txt and t4.txt files using 'time' and 'strace -c'. Copy/paste the output from the terminal window into your report.**

```
farica.mago@linux10-ea:~/palindrome$ time ./fast-pali < t4.txt
Longest palindrome: redder

real    0m0.162s
user    0m0.083s
sys     0m0.034s
farica.mago@linux10-ea:~/palindrome$ time ./fast-pali < t3.txt
Longest palindrome: ___o.O.o___

real    0m0.009s
user    0m0.003s
sys     0m0.004s
```

```
farica.mago@linux10-ea:~/palindrome$ strace -c ./fast-pali < t3.txt
Longest palindrome: ___o.O.o___
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 44.90    0.000620         620         1           execve
 27.66    0.000382          16        23           mmap
  6.66    0.000092          18         5           openat
  4.63    0.000064           9         7           mprotect
  3.48    0.000048           8         6           newfstatat
  2.82    0.000039           7         5           pread64
  2.75    0.000038           7         5           close
  2.68    0.000037           5         7           read
  1.38    0.000019          19         1         1 access
  0.87    0.000012           6         2         1 arch_prctl
  0.65    0.000009           3         3           brk
  0.58    0.000008           8         1           rseq
  0.51    0.000007           7         1           set_tid_address
  0.43    0.000006           6         1           set_robust_list
  0.00    0.000000           0         1           write
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         1           prlimit64
  0.00    0.000000           0         1           getrandom
------ ----------- ----------- --------- --------- ----------------
100.00    0.001381          19        72         2 total
farica.mago@linux10-ea:~/palindrome$ strace -c ./fast-pali < t4.txt
Longest palindrome: redder
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 95.05    0.004381         365        12           munmap
  3.93    0.000181          15        12           read
  0.52    0.000024           2        11           brk
  0.33    0.000015           0        34           mmap
  0.11    0.000005           5         1           write
  0.07    0.000003           0         6           newfstatat
  0.00    0.000000           0         5           close
  0.00    0.000000           0         7           mprotect
  0.00    0.000000           0         5           pread64
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         2         1 arch_prctl
  0.00    0.000000           0         1           set_tid_address
  0.00    0.000000           0         5           openat
  0.00    0.000000           0         1           set_robust_list
  0.00    0.000000           0         1           prlimit64
  0.00    0.000000           0         1           getrandom
  0.00    0.000000           0         1           rseq
------ ----------- ----------- --------- --------- ----------------
100.00    0.004609          43       107         2 total
farica.mago@linux10-ea:~/palindrome$
```

b) **Is your fast-pali.cpp faster than slow-pali.cpp? Why do you think that is?**

Yes, my fast-pali is faster than slow-pali.cpp. For t3.txt, fast-pali made 7 calls to the read function whereas the slow-pali made 43 calls to the read function.

For, t4.txt, fast-pali made 12 calls to the read function whereas the slow-pali made 5767198 calls to the read function.

c) **Is your program faster than palindrome.py and why?**

Yes, my program is faster than palindrome.py because,

For t3.txt, fast-pali made 7 calls to the read function whereas the palindrome.py made 80 calls to the read function.

For t4.txt, fast-pali made 12 calls to the read function whereas the palindrome.py made 784 calls to the read function.