Adarsha Kanel 30049820

Q1

a.

Palindome.py without strace -c

time python3 palindrome.py < t3.txt
Longest palindrome: ___o.O.o___

real 0m0.039s
user 0m0.022s
sys 0m0.007s

time python3 palindrome.py < t4.txt
Longest palindrome: redder

real 0m0.304s
user 0m0.300s
sys 0m0.003s

Slow-pali.cpp without strace -c

time ./slow-pali < t3.txt
Longest palindrome: ___o.O.o___

real 0m0.003s
user 0m0.000s
sys 0m0.002s

time ./slow-pali < t4.txt
Longest palindrome: redder

real 0m2.006s
user 0m0.498
sys 0m1.487s

---

b.

In t3.txt, palindrome.py spends 0.007s in kernel mode and in t4.txt, it spends 0.003s in kernel mode.
In t3.txt, palindrome.py spends 0.022s in user mode and in t4.txt, it spends 0.300s in kernel mode.

In t3.txt, slow-pali.cpp spends 0.002s in kernel mode and in t4.txt, it spends 1.487s in kernel mode.
In t3.txt, slow-pali.cpp spends 0.000s in user mode and in t4.txt, it spends 0.498s in user mode.

c.

The python runs faster in the file t4.txt than t3.txt because the slow-pali.cpp code will perform more system calls than python one as for it performs a system call for every byte read from the input text file, and thus the larger the file, the longer the sys time is.
On the other hand, because there are less system calls in smaller files as there are less characters, the slow-pali.cpp runs faster than palindrom.py file in a text file such as t3.txt.

Q3.

Fast-pali.cpp without strace -c

```
 time ./fast-pali < t3.txt
Longest palindrome: ___o.O.o___

real    0m0.003s
user    0m0.001s
sys     0m0.002s

time ./fast-pali < t4.txt
Longest palindrome: redder

real    0m0.093s
user    0m0.091s
sys     0m0.001s
```

Palindrom.py with strace -c

```
time strace -c python3 palindrome.py < t3.txt
Longest palindrome: ___o.O.o___
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 36.14    0.000850           5       151        76 openat
 11.48    0.000270           5        48           mmap
 11.22    0.000264           1       199        45 stat
  7.48    0.000176          11        16           mprotect
  6.08    0.000143           5        26           getdents
  5.95    0.000140           1       115           fstat
  5.40    0.000127           1        78           close
  4.85    0.000114           1        92           read
  3.06    0.000072          72         1           execve
  2.64    0.000062          62         1         1 access
  1.19    0.000028           0        51         4 lseek
  1.06    0.000025           0        28           brk
  0.89    0.000021           3         7           munmap
  0.68    0.000016           0        18        11 ioctl
  0.30    0.000007           0        68           rt_sigaction
  0.30    0.000007           7         1           arch_prctl
  0.21    0.000005           5         1           set_tid_address
  0.21    0.000005           5         1           set_robust_list
  0.21    0.000005           5         1           getrandom
  0.13    0.000003           3         1           rt_sigprocmask
  0.13    0.000003           1         2           futex
  0.09    0.000002           0         9           lstat
```

```
  0.09    0.000002           2         1           prlimit64
  0.04    0.000001           0         3           fcntl
  0.04    0.000001           1         1           getcwd
  0.04    0.000001           0         4         2 readlink
  0.04    0.000001           1         1           geteuid
  0.04    0.000001           1         1           getegid
  0.00    0.000000           0         1           write
  0.00    0.000000           0         3           dup
  0.00    0.000000           0         1           getpid
  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         3           sigaltstack
------ ----------- ----------- --------- --------- ----------------
100.00    0.002352                   937       139 total

real    0m0.046s
user    0m0.024s
sys     0m0.021s

time strace -c python3 palindrome.py < t4.txt
Longest palindrome: redder
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 39.75    0.000824           5       151        76 openat
 15.77    0.000327           6        48           mmap
  8.06    0.000167          10        16           mprotect
  7.43    0.000154           1       115           fstat
  6.90    0.000143           1        78           close
  6.17    0.000128           0       796           read
  5.64    0.000117           0       199        45 stat
  2.65    0.000055           0        74           brk
  0.96    0.000020           2         7           munmap
  0.92    0.000019           1        18        11 ioctl
  0.87    0.000018           0        51         4 lseek
  0.72    0.000015          15         1         1 access
  0.63    0.000013          13         1           getrandom
  0.53    0.000011           0        68           rt_sigaction
  0.53    0.000011          11         1           execve
  0.34    0.000007           7         1           arch_prctl
  0.29    0.000006           3         2           futex
  0.29    0.000006           6         1           prlimit64
  0.24    0.000005           5         1           rt_sigprocmask
  0.24    0.000005           1         4         2 readlink
```

```
  0.24    0.000005           5         1           sysinfo
  0.24    0.000005           5         1           set_tid_address
  0.24    0.000005           5         1           set_robust_list
  0.19    0.000004           1         3           dup
  0.14    0.000003           1         3           sigaltstack
  0.00    0.000000           0         1           write
  0.00    0.000000           0         9           lstat
  0.00    0.000000           0         1           getpid
  0.00    0.000000           0         3           fcntl
  0.00    0.000000           0        26           getdents
  0.00    0.000000           0         1           getcwd
  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    0.002073                  1687       139 total

real    0m0.341s
user    0m0.303s
sys     0m0.039s
```

Slow-pali.cpp with strace -c

```
time strace -c ./slow-pali < t3.txt
Longest palindrome: ___o.O.o___
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  0.00    0.000000           0        43           read
  0.00    0.000000           0         1           write
  0.00    0.000000           0         5           close
  0.00    0.000000           0         8         7 stat
  0.00    0.000000           0         6           fstat
  0.00    0.000000           0        14           mmap
  0.00    0.000000           0        10           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         1           arch_prctl
  0.00    0.000000           0        48        43 openat
------ ----------- ----------- --------- --------- ----------------
100.00    0.000000                   142        51 total
```

```
real    0m0.019s
user    0m0.007s
sys     0m0.004s


time strace -c ./slow-pali < t4.txt
Longest palindrome: redder
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
100.00   11.705803           2   5767198           read
  0.00    0.000012           1        10           mprotect
  0.00    0.000009           9         1           munmap
  0.00    0.000007           7         1           write
  0.00    0.000005           1         3           brk
  0.00    0.000004           0        14           mmap
  0.00    0.000002           0         6           fstat
  0.00    0.000000           0         5           close
  0.00    0.000000           0         8         7 stat
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         1           arch_prctl
  0.00    0.000000           0        48        43 openat
------ ----------- ----------- --------- --------- ----------------
100.00   11.705842             5767297        51 total

real    0m54.293s
user    0m8.162s
sys     0m56.553s
```

Fast-pali.cpp with strace -c

```
 time strace -c ./fast-pali < t3.txt
Longest palindrome: ___o.O.o___
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 26.96    0.000093           9        10           mprotect
 22.61    0.000078           1        48        43 openat
 18.55    0.000064           4        14           mmap
  6.67    0.000023           3         6           read
  6.67    0.000023          23         1           munmap
  5.22    0.000018          18         1           write
  4.35    0.000015           5         3           brk
  3.77    0.000013           2         6           fstat
  3.48    0.000012           2         5           close
  1.74    0.000006           6         1           arch_prctl
```

```
  0.00    0.000000           0         8         7 stat
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
------ ----------- ----------- --------- --------- ----------------
100.00    0.000345                   105        51 total

real    0m0.014s
user    0m0.003s
sys     0m0.008s

 time strace -c ./fast-pali < t4.txt
Longest palindrome: redder
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 49.49    0.000828          75        11           read
 21.82    0.000365           7        48        43 openat
  7.65    0.000128           9        14           mmap
  4.72    0.000079           7        10           mprotect
  3.83    0.000064          64         1         1 access
  3.65    0.000061          20         3           brk
  3.29    0.000055           6         8         7 stat
  1.97    0.000033          33         1           execve
  1.67    0.000028           4         6           fstat
  1.61    0.000027           5         5           close
  0.30    0.000005           5         1           arch_prctl
  0.00    0.000000           0         1           write
  0.00    0.000000           0         1           munmap
------ ----------- ----------- --------- --------- ----------------
100.00    0.001673                   110        51 total

real    0m0.112s
user    0m0.091s
sys     0m0.011s
```

My fast-pali.cpp is faster than the slow-pali.cpp because it performs less system calls, 105 for t3.txt and 110 for t4.txt when compared to slow-pali.cpp which does 142 system call for t3.txt and 5767297 system call for t4.txt. So, because slow-pali.cpp has to do more system calls for the same result as fast-pali.cpp, it has to spend much more time in kernel mode, slowing down the code significantly.

B.

The code fast-pali.cpp is faster than python because fast-pali.cpp performs less system calls, 105 for t3.txt and 110 for t4.txt than palindrom.py, which has 937 for t3.txt and 1687 for t4.txt. Thus, because of this fast-pali.cpp stayed less time in sys mode, 0.008s in t3.txt, and 0.001s in t4.txt, than palindrome.py, which spent 0.007s in t3.txt and 0.003 in t4.txt, making it faster than palindrome.py. Also, because python is an interpreted language, it will be compiled while running, on the other hand c++ is a compiled language, and is compiled before it is run, making python run slower than c++ as it will have to run and compile the code when it's running.