CPSC 457 Operating Systems

Talha Khalil, 30037871

Assignment 1

Question 1

A

```
talha.khalil1@linux02-wc:~/OperatingSystems/Assignment_1/palindrome$ time ./palindrome.py < t4.txt
Longest palindrome: redder

real    0m0.241s
user    0m0.228s
sys     0m0.010s
```

```
talha.khalil1@linux02-wc:~/OperatingSystems/Assignment_1/palindrome$ time ./palindrome.py < t3.txt
Longest palindrome: ___o.O.o___

real    0m0.038s
user    0m0.021s
sys     0m0.012s
```

```
talha.khalil1@linux02-wc:~/OperatingSystems/Assignment_1/palindrome$ time ./slow-pali < t4.txt
Longest palindrome: redder

real    0m1.124s
user    0m0.200s
sys     0m0.920s
```

```
talha.khalil1@linux02-wc:~/OperatingSystems/Assignment_1/palindrome$ time ./slow-pali < t3.txt
Longest palindrome: ___o.O.o___

real    0m0.006s
user    0m0.001s
sys     0m0.004s
```

B

User time is user…

Kernel time is real-user

i. Palindrome.py < t4.txt spent 0.228 s in user mode, and 0.013s in kernel mode
ii. Palindrome.py < t3.txt spent 0.021 s in user mode, and 0.017s in kernel mode
iii. Slow-pali.cpp < t4.txt spent 0.200 s in user mode, and 0.924s in kernel mode
iv. Slow-pali.cpp < t3.txt spent 0.001 s in user mode, and 0.005s in kernel mode

C

```
talha.khalil1@linux02-wc:~/OperatingSystems/Assignment_1/palindrome$ strace -c ./palindrome.py < t4.txt
Longest palindrome: redder
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 22.83    0.000872           1       496        56 newfstatat
 21.29    0.000813           2       373       245 openat
 10.74    0.000410           2       174           mmap
 10.29    0.000393           0       869           read
 10.19    0.000389          77         5         3 execve
  4.63    0.000177           1       132           close
  3.59    0.000137           2        48           mprotect
  3.40    0.000130           1       127         2 lseek
  3.19    0.000122           5        24           getdents64
  2.83    0.000108           1        68           rt_sigaction
  1.99    0.000076           1        71        67 ioctl
  1.28    0.000049           6         8           munmap
  0.71    0.000027           0        28           brk
  0.52    0.000020           2        10           pread64
  0.47    0.000018           3         5         4 access
  0.29    0.000011           2         4         3 readlink
  0.29    0.000011           1         6         4 prctl
  0.24    0.000009           4         2         2 statfs
  0.24    0.000009           2         4         2 arch_prctl
  0.16    0.000006           2         3           dup
  0.16    0.000006           2         3           fcntl
  0.13    0.000005           2         2           getcwd
  0.08    0.000003           3         1           gettid
  0.05    0.000002           2         1           rt_sigprocmask
  0.05    0.000002           1         2           futex
  0.05    0.000002           2         1           set_tid_address
  0.05    0.000002           2         1           set_robust_list
  0.05    0.000002           2         1           epoll_create1
  0.05    0.000002           2         1           prlimit64
  0.05    0.000002           2         1           getrandom
  0.03    0.000001           1         1           getuid
  0.03    0.000001           1         1           getgid
  0.03    0.000001           1         1           geteuid
  0.03    0.000001           1         1           getegid
  0.00    0.000000           0         1           write
  0.00    0.000000           0         1           uname
  0.00    0.000000           0         1           sysinfo
------ ----------- ----------- --------- --------- ----------------
100.00    0.003819           1      2478       388 total
```

```
talha.khalil1@linux02-wc:~/OperatingSystems/Assignment_1/palindrome$ strace -c ./palindrome.py < t3.txt
Longest palindrome: ___o.O.o___
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 32.91    0.000666           1       373       245 openat
 18.23    0.000369           2       174           mmap
```

```
 15.12    0.000306           0       496        56 newfstatat
  5.58    0.000113           0       132           close
  5.24    0.000106          21         5         3 execve
  5.19    0.000105           0       165           read
  5.19    0.000105           4        24           getdents64
  4.94    0.000100           2        48           mprotect
  1.33    0.000027           0       127         2 lseek
  1.09    0.000022           1        17           brk
  1.09    0.000022           2        10           pread64
  0.89    0.000018           2         8           munmap
  0.69    0.000014           0        71        67 ioctl
  0.35    0.000007           1         5         4 access
  0.30    0.000006           1         4         2 arch_prctl
  0.25    0.000005           1         4         3 readlink
  0.25    0.000005           5         1           epoll_create1
  0.20    0.000004           0        68           rt_sigaction
  0.15    0.000003           1         2           getcwd
  0.15    0.000003           3         1           gettid
  0.10    0.000002           2         1           rt_sigprocmask
  0.10    0.000002           2         1           sysinfo
  0.10    0.000002           0         6         4 prctl
  0.10    0.000002           1         2           futex
  0.10    0.000002           2         1           set_tid_address
  0.10    0.000002           2         1           set_robust_list
  0.10    0.000002           2         1           prlimit64
  0.10    0.000002           2         1           getrandom
  0.05    0.000001           1         1           uname
  0.05    0.000001           0         2         2 statfs
  0.00    0.000000           0         1           write
  0.00    0.000000           0         3           dup
  0.00    0.000000           0         3           fcntl
  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    0.002024           1      1763       388 total
```

```
talha.khalil1@linux02-wc:~/OperatingSystems/Assignment_1/palindrome$ strace -c ./slow-pali < t4.txt
Longest palindrome: redder
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
100.00   11.349926           1   5767198           read
  0.00    0.000005           5         1           write
  0.00    0.000005           0        16         9 newfstatat
  0.00    0.000000           0         5           close
  0.00    0.000000           0        22           mmap
  0.00    0.000000           0         9           mprotect
  0.00    0.000000           0         1           munmap
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         4           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        58        53 openat
------ ----------- ----------- --------- --------- ----------------
100.00   11.349936           1   5767321        64 total
```

```
talha.khalil1@linux02-wc:~/OperatingSystems/Assignment_1/palindrome$ strace -c ./slow-pali < t3.txt
Longest palindrome: ___o.O.o___
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 42.00    0.000147           2        58        53 openat
 19.43    0.000068           3        22           mmap
 14.57    0.000051           1        43           read
 10.86    0.000038           2        16         9 newfstatat
  6.00    0.000021           2         9           mprotect
  2.29    0.000008           1         5           close
  2.29    0.000008           2         4           pread64
  0.86    0.000003           3         1           munmap
  0.57    0.000002           2         1           write
  0.57    0.000002           0         3           brk
  0.57    0.000002           1         2         1 arch_prctl
  0.00    0.000000           0         1         1 access
  0.00    0.000000           0         1           execve
------ ----------- ----------- --------- --------- ----------------
100.00    0.000350           2       166        64 total
```

D

Comparing the amount of system calls explains why sometimes the python code is faster and sometimes slower, in the case of t4.txt python makes 2478 system calls where as slow-plai makes 5767321 calls. Whereas for t3.txt python makes 1763 calls, and slow-pali makes 166. System calls are expensive therefore, having more of them slows down a program.

Question 3

A

```
talha.khalil1@linux02-wc:~/OperatingSystems/Assignment_1$ time -p ./fast-pali < t3.txt
Longest palindrome: ___o.O.o___
real 0.00
user 0.00
sys 0.00
```

```
talha.khalil1@linux02-wc:~/OperatingSystems/Assignment_1$ time -p ./fast-pali < t4.txt
Longest palindrome: redder
real 0.03
user 0.03
sys 0.00
```

```
talha.khalil1@linux02-wc:~/OperatingSystems/Assignment_1$ strace -c ./fast-pali < t3.txt
Longest palindrome: ___o.O.o___
```

```
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 27.64    0.000149           2        58        53 openat
 25.60    0.000138         138         1           execve
 23.01    0.000124           5        22           mmap
  6.86    0.000037           2        16         9 newfstatat
  6.12    0.000033           3         9           mprotect
  2.60    0.000014           2         6           read
  1.86    0.000010           2         5           close
  1.48    0.000008           2         4           pread64
  1.30    0.000007           7         1           munmap
  1.11    0.000006           2         3           brk
  0.93    0.000005           5         1           write
  0.74    0.000004           4         1         1 access
  0.74    0.000004           2         2         1 arch_prctl
------ ----------- ----------- --------- --------- ----------------
100.00    0.000539           4       129        64 total
```

```
talha.khalil1@linux02-wc:~/OperatingSystems/Assignment_1$ strace -c ./fast-pali < t4.txt
Longest palindrome: redder
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 96.25    0.000949         118         8           read
  2.13    0.000021           2         9           mprotect
  0.81    0.000008           8         1           munmap
  0.51    0.000005           1         3           brk
  0.30    0.000003           0        22           mmap
  0.00    0.000000           0         1           write
  0.00    0.000000           0         5           close
  0.00    0.000000           0         4           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        58        53 openat
  0.00    0.000000           0        16         9 newfstatat
------ ----------- ----------- --------- --------- ----------------
100.00    0.000986           7       131        64 total
```

B

Yes, it is. This is because of the amount read calls that are made. Fast-pali makes significantly less sys calls.

C

Yes, it is. This is because for two reasons, one my program makes fewer read calls, and two that python has an inherent overhead because it's an interpreted language where it has to make more system calls before even executing the user-written code. Meaning python would have a fixed overhead, whereas the fast-pali does not.