Arda Kıray
21302072
Section 2

# CS 342 - Operating Systems
# Project 1

**Question 3 - Part A :**

I created a FileCreator program as explained in the README.txt file. After checking the validity of the programs with various files that I created, by using FileCreator, I created a randomly constructed files with 50000 entries. I tested my phistogram and thistogram programs by using these files and my findings in terms of run-time are below :

Some conventions:
Real means the real time elapsed in terms of universal time like clock time.
User means the time elapsed as CPU seconds in the user mode.
Sys means the time elapsed as CPU seconds in the kernel mode.
The timing format is *m*.***s, the asterisk number before m represents minutes and the number before s represents the seconds.

**Phistogram :**

**1st Iteration :**

| For 1 file : | | For 2 files : | | For 4 files : | | For 8 files : | |
|---|---|---|---|---|---|---|---|
| real | 0m0.010s. | real | 0m0.010s | real | 0m0.025s | real | 0m0.055s |
| user | 0m0.009s | user | 0m0.017s | user | 0m0.026s | user | 0m0.070s |
| sys | 0m0.000s | sys | 0m0.000s | sys | 0m0.010s | sys. | 0m0.000s |

**2nd Iteration :**

| For 1 file : | | For 2 files : | | For 4 files : | | For 8 files : | |
|---|---|---|---|---|---|---|---|
| real | 0m0.017s. | real | 0m0.011s | real | 0m0.020s | real | 0m0.044s |
| user | 0m0.010s | user | 0m0.017s | user | 0m0.032s | user | 0m0.065s |
| sys | 0m0.004s | sys | 0m0.000s | sys | 0m0.000s | sys. | 0m0.004s |

**3rd Iteration :**

| For 1 file : | | For 2 files : | | For 4 files : | | For 8 files : | |
|---|---|---|---|---|---|---|---|
| real | 0m0.010s. | real | 0m0.011s | real | 0m0.029s | real | 0m0.051s |
| user | 0m0.009s | user | 0m0.017s | user | 0m0.042s | user | 0m0.040s |
| sys | 0m0.000s | sys | 0m0.000s | sys | 0m0.010s | sys. | 0m0.030s |

**4th Iteration :**

| For 1 file : | | For 2 files : | | For 4 files : | | For 8 files : | |
|---|---|---|---|---|---|---|---|
| real | 0m0.010s. | real | 0m0.022s | real | 0m0.020s | real | 0m0.045s |
| user | 0m0.005s | user | 0m0.026s | user | 0m0.033s | user | 0m0.056s |
| sys | 0m0.004s | sys | 0m0.007s | sys | 0m0.001s | sys. | 0m0.020s |

**Results :**

**The mean values:**

| For 1 file : | | For 2 files : | | For 4 files : | | For 8 files : | |
|---|---|---|---|---|---|---|---|
| real | 0.01175s. | real | 0.01350s | real | 0.02350s | real | 0.04875s |
| user | 0.00825s | user | 0.01925s | user | 0.03325s | user | 0.05775s |
| sys | 0.00200s | sys | 0.00175s | sys | 0.00525s | sys | 0.01350s |

For 1 file :
- Standard deviation of real time is 0.0035 seconds and variance is 0.01225 seconds.
- Standard deviation of user time is 0.0022 seconds and variance is 0.00491 seconds.
- Standard deviation of system time is 0.0023 seconds and variance is 0.00533 seconds.

For 2 files :
- Standard deviation of real time is 0.0057 seconds and variance is 0.03233 seconds.
- Standard deviation of user time is 0.0045 seconds and variance is 0.02025 seconds.
- Standard deviation of system time is 0.0035 seconds and variance is 0.01225 seconds.

For 4 files :
- Standard deviation of real time is 0.0043 seconds and variance is 0.01900 seconds.
- Standard deviation of user time is 0.0066 seconds and variance is 0.04558 seconds.
- Standard deviation of system time is 0.0055 seconds and variance is 0.03025 seconds.

For 8 files :
- Standard deviation of real time is 0.0052 seconds and variance is 0.02892 seconds.
- Standard deviation of user time is 0.0132 seconds and variance is 0.17359 seconds.
- Standard deviation of system time is 0.0140 seconds and variance is 0.19566 seconds.

The mean runtime values for real, user and system times are increasing with the number of the files processed with one error which is the system time value decreases for 2 files compared to 1 file. As written in the manual of the time command of Linux, since time is not collected atomically with the execution of the program, some errors may occurs. For example, if the running commands' CPU usage percentage is very close to zero, they may be reported as zero. The mean running times of the program increases nearly linearly with the given number of files. Since phistogram is not multi-threaded, the processes are not running concurrently. Therefore, the amount of time it takes to finish the whole work increases in a linear-like fashion.

**Thistogram :**

**1st Iteration :**

| For 1 file : | | For 2 files : | | For 4 files : | | For 8 files : | |
|---|---|---|---|---|---|---|---|
| real | 0m0.011s. | real | 0m0.013s | real | 0m0.029s | real | 0m0.052s |
| user | 0m0.011s | user | 0m0.023s | user | 0m0.052s | user | 0m0.080s |
| sys | 0m0.000s | sys | 0m0.000s | sys | 0m0.000s | sys. | 0m0.008s |

**2nd Iteration :**

| For 1 file : | For 2 files : | For 4 files : | For 8 files : |
|---|---|---|---|
| real    0m0.013s. | real    0m0.017s | real    0m0.029s | real   0m0.053s |
| user    0m0.012s | user    0m0.028s | user    0m0.047s | user 0m0.073s |
| sys    0m0.000s | sys    0m0.000s | sys    0m0.004s | sys.  0m0.019s |

**3rd Iteration :**

| For 1 file : | For 2 files : | For 4 files : | For 8 files : |
|---|---|---|---|
| real    0m0.014s. | real    0m0.013s | real    0m0.027s | real   0m0.055s |
| user    0m0.012s | user    0m0.023s | user    0m0.036s | user 0m0.090s |
| sys    0m0.000s | sys    0m0.000s | sys    0m0.009s | sys.  0m0.004s |

**4th Iteration :**

| For 1 file : | For 2 files : | For 4 files : | For 8 files : |
|---|---|---|---|
| real    0m0.014s. | real    0m0.013s | real    0m0.024s | real   0m0.055s |
| user    0m0.009s | user    0m0.018s | user    0m0.037s | user 0m0.092s |
| sys    0m0.003s | sys    0m0.005s | sys    0m0.004s | sys.  0m0.004s |

**Results :**

**The mean values:**

| For 1 file : | For 2 files : | For 4 files : | For 8 files : |
|---|---|---|---|
| real    0.01300s. | real    0.01400s | real    0.02725s | real   0.05375s |
| user    0.01100s | user    0.02300s | user    0.04300s | user 0.08375s |
| sys    0.00075s | sys    0.00125s | sys    0.00425s | sys   0.00875s |

For 1 file :
- Standard deviation of real time is 0.0014 seconds and variance is 0.00200 seconds.
- Standard deviation of user time is 0.0014 seconds and variance is 0.00200 seconds.
- Standard deviation of system time is 0.0015 seconds and variance is 0.00225 seconds.

For 2 files :
- Standard deviation of real time is 0.0020 seconds and variance is 0.00400 seconds.
- Standard deviation of user time is 0.0041 seconds and variance is 0.01666 seconds.
- Standard deviation of system time is 0.0025 seconds and variance is 0.00625 seconds.

For 4 files :
- Standard deviation of real time is 0.0024 seconds and variance is 0.00558 seconds.
- Standard deviation of user time is 0.0078 seconds and variance is 0.06066 seconds.
- Standard deviation of system time is 0.0037 seconds and variance is 0.01358 seconds.

For 8 files :
- Standard deviation of real time is 0.0015 seconds and variance is 0.00225 seconds.
- Standard deviation of user time is 0.0089 seconds and variance is 0.07892 seconds.
- Standard deviation of system time is 0.0071 seconds and variance is 0.05025 seconds.

The mean runtime values for multi-threaded thistogram application still increases very close to linearly in terms of the input file number. The reason is that even if the program is designed to create one thread for each file, the CPU that my computer has which is Intel i5 core with 2 processing cores, the two cores shares the same FPU(Floating-point unit). The thistogram program takes the inputs as floating point values to be able to process both integers and floating point numbers. However, since the FPU is shared among the processing cores, thistogram cannot run the threads fully concurrently on my machine. Therefore, the resulting run-time values are very close to the phistogram program. Moreover, since the processor consists of 2 cores, only two threads can be executed at a time instant, therefore as can be seen in the mean results the jump between 1 file to 2 files is smaller compared to jump between 2 to 4 files since the maximum amount of threads can be executed concurrently is 2.

**Question 3 - Part B :**

**Phistogram :**

**1st Iteration :** Input size is 10.000, input range is [0,50000], number of bins are 50

```
real    0m0.012s
user    0m0.011s
sys     0m0.003s
```

**2nd Iteration :** Input size is 100.000, input range is [0,50000], number of bins are 50

```
real    0m0.027s
user    0m0.033s
sys     0m0.013s
```

**3rd Iteration :** Input size is 1.000.000, input range is [0,50000], number of bins are 50

```
real    0m0.231s
user    0m0.301s
sys     0m0.021s
```

**4th Iteration :** Input size is 10.000.000, input range is [0,50000], number of bins are 50

```
real    0m2.914s
user    0m2.791s
sys     0m0.227s
```

**5th Iteration :** Input size is 100.000.000, input range is [0,50000], number of bins are 50

```
real    0m30.494s
user    0m29.673s
sys     0m2.146s
```

The input size of the programs multiplies with 10 for each next iteration. As can be seen, the run-time values are nearly multiplied by 10 also. Therefore, the linear increment can also be seen from this experiment.