

CS 3360 Programming Assignment 1

Andrew Bracken

Submitted Files:

- Source Code:
 - Problem_1.cpp
 - Problem_2.cpp
- README text file

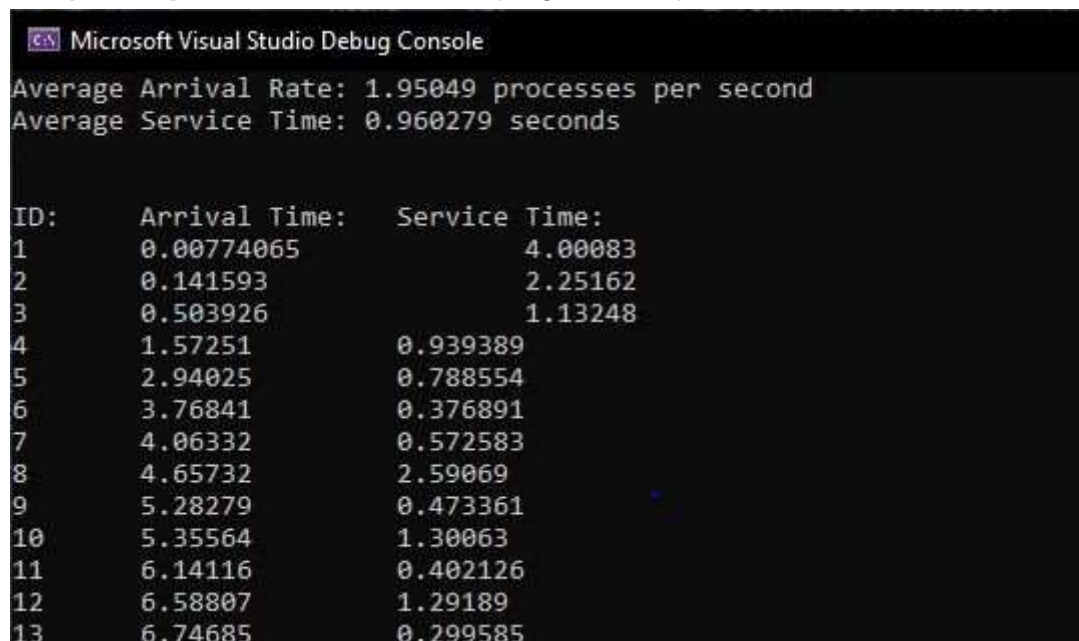
Problem 1: Problem_1.cpp

For this problem, I wrote a C++ program to simulate a system's arrival and service of processes using random arrival rates, following a Poisson-distribution, and exponentially distributed service times. Additionally, the program calculates the actual average arrival rate and service time of the simulated system.

Average Arrival Rate: 1.95049 processes per second

Average Service Time: 0.960279

Sample Output: Small section of the program's output.



The screenshot shows the Microsoft Visual Studio Debug Console with the following output:

```
Microsoft Visual Studio Debug Console
Average Arrival Rate: 1.95049 processes per second
Average Service Time: 0.960279 seconds

ID:      Arrival Time:      Service Time:
1        0.00774065          4.00083
2        0.141593            2.25162
3        0.503926            1.13248
4        1.57251             0.939389
5        2.94025             0.788554
6        3.76841             0.376891
7        4.06332             0.572583
8        4.65732             2.59069
9        5.28279             0.473361
10       5.35564             1.30063
11       6.14116             0.402126
12       6.58807             1.29189
13       6.74685             0.299585
```

Problem2: Problem_2.cpp

For this problem, I wrote a C++ program to generate synthetic data for the servers failure and restoration times over a twenty year period, using randomly generated uptimes based on an MTBF of 500 hours and following an exponential distribution.

Sample Output:

```
Microsoft Visual Studio Debug Console

System-wide failure detected at: 4846.51
Server 1 failed at: 4846.51
Server 2 failed at: 4840.17

Server 1 Failure:      Restored:
748.984                758.984
2607.65                2617.65
2988.84                2998.84
3351.29                3361.29
4021.6                 4031.6
4846.51                4856.51
5001.81                5011.81
5870.72                5880.72
6905.36                6915.36
7004.33                7014.33
7524.71                7534.71
8042.14                8052.14
9137.42                9147.42
9268.64                9278.64
10027.8                10037.8
```

```
173741                173751
174051                174061
174617                174627

Server 2 Failure:      Restored:
558.092                568.092
980.953                990.953
1756.19                1766.19
2049.82                2059.82
2442.31                2452.31
3063.01                3073.01
3196.39                3206.39
3421.07                3431.07
3609.34                3619.34
4840.17                4850.17
5156.08                5166.08
5340                  5350
6350.08                6360.08
6365.14                6375.14
```

To find an average time until a system-wide failure occurred, I ran the program 10 times and computed a rough average below:

$(4846.5 + 3443.05 + 6538.94 + 7631.19 + 1454.68 + 3119.81 + 2089.06 + 1644.42 + 4368.79 + 8465.22) / 10 = \text{system wide failure every } 4,333.166 \text{ hours}$