CS 342 Project 1
Yasin Balcancı
21501109

Part A)

We give the parameters minvalue, maxvalue, bincount, n, file1…fileN, output to this program from argv while running it from the terminal. It creates intermediate files according to the names of input files. For example input files are named “file1” and “file2”, intermediate output files will be named “out-file1” and “out-file2”. Name of the ultimate output file must be provided as a parameter.

Part B)

We give the parameters minvalue, maxvalue, bincount, n, file1…fileN, output to this program from argv while running it from the terminal. It keeps minval, maxval, bincount, n and an array representing the bins as global variables. Then in main function, the values in the array are written to the output file.

Part C)

a)

Given inputs:

For 1 thread/process: 1 file consisting of 32 numbers.
For 2 thread/process: 2 file each consisting of 16 numbers.
For 4 thread/process: 4 file each consisting of 8 numbers.
For 8 thread/process: 8 file each consisting of 4 numbers.

Table 1: Tests for multithread application(MS)

Threads	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Avg.
1	1214	1347	1416	1660	1424	1469	1519	1436
2	1635	1929	1824	1794	1630	1407	1587	1686
4	2372	1899	1865	1879	2018	1767	2173	1996
8	4047	2019	2428	2477	2645	1965	1952	2504

Table 2: Tests for multiprocess application(MS)

Threads	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Avg.
1	1864	1569	1681	1719	1739	1546	1719	1691
2	723	2153	1742	3317	1756	1793	1656	1877
4	1120	1521	2232	1285	1736	2635	2352	1840
8	1290	1958	2042	2211	1883	1903	1846	1876

Time required to process the files seem to increase as the number of threads increase. But the time required did not change significantly as the number of processes increase.