HW2: Process Scheduling Algorithms

Team 7

Jonathan Weinstein 009789039

…

We implemented this assignment using C++11, though most of the code can pass as plain C.

To run our program using GCC or compatible compiler, place all source and header files in the same directory and issue:

g++ -std=c++11 -Wall -Wextra \*.cpp

All 6 of the required algorithms are completed, as well as the two extra-credit (Aging variants of HPF, both preemptive/not) for a total of 8.

There is test function that verifies the frequency of time slices and a jobs total burst. Algorithm outputs were also examined and checked against edge cases.

The structure of main is an outermost loop that iterates for each algorithm, stored in a function pointer. All algorithms have the same signature. It is important to note that each algorithm runs on 5 data sets, before moving onto the next algorithm. The random number generator is reseeded to the same value before test 0 of each algorithm, so all algorithms work on the same data.

The algorithm signature contains an output array of

struct PerJobStats

**{**

int qbegin**,** qend**;**

**};**

Indexed by job ID. qbegin is the quanta when the job got its first time slice, and qend is when it finished (note this range is exclusive, see stats below). This schema makes the implementation of each algorithm simpler, instead of them all computing the 3 stats. These stats are computed after the algorithm runs, before they are printed on the same line.

int const response **=** stats**[**i**].**qbegin **-** job**[**i**].**arrival**;**

int const turnaround **=** stats**[**i**].**qend **-** job**[**i**].**arrival**;**

int const wait **=** turnaround **-** job**[**i**].**burst**;**

Jonathan observed that SJF, HPF(no preempt), and HPF-Aging(non-preempt) are all very similar to implement (same for their preemptive versions). They store pending jobs in a collection, ordered by some heuristic. They take the front most job and run it.

The ordering of the collection based on a comparison on an integer field. The only thing different is the field compared on!

SJF compares on remaining burst time.

HPF compares on priority.

HPF-Aging also compares on "priority," but I slyly store (initial\_priority\*5 + arrival time) in that field. Only a jobs priority relative to another is of interest.

(promoted 1 priority every 5 secs, and + because lower priority value more important)

So I knocked out all three with the following template function, which allows for a different comparison function (or C++ ‘functor’) and what data to insert when inserting a job into the collection (the Aging parameter).

dat**.**priority **=** aging **?** jb**.**priority**\***5u **+** jb**.**arrival **:** jb**.**priority**;**

template**<**class CmpFunc**,** bool Aging**=false>**

AlgoRet non\_preempt**(**const Job **\***job**,** int njobs**,** PerJobStats **\***stats**,** char **\***gantt**);**

For example, SJF is:

non\_preempt**<**SrtComp**,** **false>**

**Stat results and analysis:**

The tail of this document contains the program output. Once again, an algorithm runs on each data set (test 0 to 4) before moving to the next algorithm. Wait time, turnaround, and response are displayed after each test. So is a gantt chart, showing the ID of what process ran during that quanta, ‘.’ Indicates the cpu was idle. After all 5 tests run, averages are displayed, and the next algorithm begins.

Analysis:

First Come First Serve is simple to implement with few context switches. Wait times can very a lot due to high burst time processes clogging the queue, making it unsuitable for a user-device. However, globally it may be a good choice in situations where wait time is not an issue.

Shortest Job First and Shortest Remaining Time both have very optimized performance across the board. For our simulation purposes, these

algorithms would be the best. However, these two rely on the CPU having prior knowledge of the burst times of the processes they are

supposed to service. Therefore, these two algorithms have limited application.

Round Robin has the most optimized response time because time is evenly distributed between the processes, giving all of them

a chance to run making it an excellent choice for a user-device. It is somewhat complicated to implement, especially as a ‘true’ round robin, and its many context switches would make it a poor choice globally and not suitable for a database.

Although the Priority Algorithms seem to perform poorly, their advantage is not captured through this simulation due to all processes

getting assigned random priorities. Presumably in the real world, processes with higher priorities are more important. These leads

to cases of inefficiency if a high priority process has a large burst time that clogs up the priority queue. This problem is fixed

by adding aging, which raises the priority of processes that have not been serviced in a long time. Aging appears to improve performance

of priority algorithms in all respects.

**NOTE:**

This output to be printed was ran by overriding the number of data sets to 2 instead of 5 as a cmdline argument. The uploaded document will contain all 5. This is to reduce the pages printed to 14 from 28.

C:\Users\jw\Desktop\spring18\opsys\os\hw2>ssim.exe 2

Seed: 0xCAFEBEEF, Number of tests: 2

\*\*\* Testing algorithm: First Come First Serve \*\*\*

Test no: 0

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 2 : 0 0 16

B 13 5 4 : 3 3 8

C 17 13 4 : 4 4 17

D 31 9 3 : 3 3 12

E 38 4 4 : 5 5 9

F 39 12 1 : 8 8 20

G 53 14 1 : 6 6 20

H 68 8 1 : 5 5 13

I 75 13 4 : 6 6 19

J 81 5 2 : 13 13 18

K 86 5 4 : 13 13 18

L 93 14 2 : ~ ~ ~

M 106 3 2 : ~ ~ ~

N 107 15 4 : ~ ~ ~

Average wait : 66/11 = 6.000

Average response : 66/11 = 6.000

Average turnaround : 170/11 = 15.455

Throughput: 11/104 = 0.105769 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAABBBBBCCCCCCCCCCCCCDDDDDDDDDEEEEFFFFFFFFFFFFGGGGGGGGGGGGGGHHHHHHHHIIIIIIIIIIIIIJJJJJKKKKK

Test no: 1

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 3 : 0 0 16

B 17 3 4 : 0 0 3

C 22 8 2 : 0 0 8

D 28 11 4 : 2 2 13

E 36 9 3 : 5 5 14

F 39 16 4 : 11 11 27

G 55 5 2 : 11 11 16

H 62 12 2 : 9 9 21

I 68 13 3 : 15 15 28

J 82 9 2 : 14 14 23

K 83 15 1 : ~ ~ ~

L 90 10 4 : ~ ~ ~

M 98 5 4 : ~ ~ ~

N 100 5 2 : ~ ~ ~

Average wait : 67/10 = 6.700

Average response : 67/10 = 6.700

Average turnaround : 169/10 = 16.900

Throughput: 10/105 = 0.095238 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAA.BBB..CCCCCCCCDDDDDDDDDDDEEEEEEEEEFFFFFFFFFFFFFFFFGGGGGHHHHHHHHHHHHIIIIIIIIIIIIIJJJJJJJJJ

All 2 tests for [First Come First Serve] done, averages:

Wait : 6.350

Response : 6.350

Turnaround: 16.177

Throughput: 10.050 per 100 quanta

\*\*\* Testing algorithm: Shortest Job First (non-preemptive) \*\*\*

Test no: 0

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 2 : 0 0 16

B 13 5 4 : 3 3 8

C 17 13 4 : 4 4 17

D 31 9 3 : 3 3 12

E 38 4 4 : 5 5 9

F 39 12 1 : 8 8 20

G 53 14 1 : 6 6 20

H 68 8 1 : 5 5 13

I 75 13 4 : 16 16 29

J 81 5 2 : 0 0 5

K 86 5 4 : 0 0 5

L 93 14 2 : ~ ~ ~

M 106 3 2 : ~ ~ ~

N 107 15 4 : ~ ~ ~

Average wait : 50/11 = 4.545

Average response : 50/11 = 4.545

Average turnaround : 154/11 = 14.000

Throughput: 11/104 = 0.105769 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAABBBBBCCCCCCCCCCCCCDDDDDDDDDEEEEFFFFFFFFFFFFGGGGGGGGGGGGGGHHHHHHHHJJJJJKKKKKIIIIIIIIIIIII

Test no: 1

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 3 : 0 0 16

B 17 3 4 : 0 0 3

C 22 8 2 : 0 0 8

D 28 11 4 : 2 2 13

E 36 9 3 : 5 5 14

F 39 16 4 : 11 11 27

G 55 5 2 : 11 11 16

H 62 12 2 : 9 9 21

I 68 13 3 : ~ ~ ~

J 82 9 2 : 1 1 10

K 83 15 1 : ~ ~ ~

L 90 10 4 : 2 2 12

M 98 5 4 : ~ ~ ~

N 100 5 2 : ~ ~ ~

Average wait : 41/10 = 4.100

Average response : 41/10 = 4.100

Average turnaround : 140/10 = 14.000

Throughput: 10/102 = 0.098039 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAA.BBB..CCCCCCCCDDDDDDDDDDDEEEEEEEEEFFFFFFFFFFFFFFFFGGGGGHHHHHHHHHHHHJJJJJJJJJLLLLLLLLLL

All 2 tests for [Shortest Job First (non-preemptive)] done, averages:

Wait : 4.323

Response : 4.323

Turnaround: 14.000

Throughput: 10.190 per 100 quanta

\*\*\* Testing algorithm: Shortest Remaining Time (preemptive) \*\*\*

Test no: 0

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 2 : 0 0 16

B 13 5 4 : 3 3 8

C 17 13 4 : 4 4 17

D 31 9 3 : 3 7 16

E 38 4 4 : 0 0 4

F 39 12 1 : 8 8 20

G 53 14 1 : 6 6 20

H 68 8 1 : 5 5 13

I 75 13 4 : 16 16 29

J 81 5 2 : 0 0 5

K 86 5 4 : 0 0 5

L 93 14 2 : 11 14 28

M 106 3 2 : 0 0 3

N 107 15 4 : ~ ~ ~

Average wait : 63/13 = 4.846

Average response : 56/13 = 4.308

Average turnaround : 184/13 = 14.154

Throughput: 13/121 = 0.107438 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAABBBBBCCCCCCCCCCCCCDDDDEEEEDDDDDFFFFFFFFFFFFGGGGGGGGGGGGGGHHHHHHHHJJJJJKKKKKIIIIIIIIIIIIILLMMMLLLLLLLLLLLL

Test no: 1

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 3 : 0 0 16

B 17 3 4 : 0 0 3

C 22 8 2 : 0 0 8

D 28 11 4 : 2 2 13

E 36 9 3 : 5 5 14

F 39 16 4 : 11 16 32

G 55 5 2 : 0 0 5

H 62 12 2 : 9 9 21

I 68 13 3 : ~ ~ ~

J 82 9 2 : 1 1 10

K 83 15 1 : ~ ~ ~

L 90 10 4 : 2 2 12

M 98 5 4 : ~ ~ ~

N 100 5 2 : ~ ~ ~

Average wait : 35/10 = 3.500

Average response : 30/10 = 3.000

Average turnaround : 134/10 = 13.400

Throughput: 10/102 = 0.098039 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAA.BBB..CCCCCCCCDDDDDDDDDDDEEEEEEEEEFFFFFGGGGGFFFFFFFFFFFHHHHHHHHHHHHJJJJJJJJJLLLLLLLLLL

All 2 tests for [Shortest Remaining Time (preemptive)] done, averages:

Wait : 4.173

Response : 3.654

Turnaround: 13.777

Throughput: 10.274 per 100 quanta

\*\*\* Testing algorithm: Round Robin \*\*\*

Test no: 0

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 2 : 0 2 18

B 13 5 4 : 1 5 10

C 17 13 4 : 2 6 19

D 31 9 3 : 1 10 19

E 38 4 4 : 1 7 11

F 39 12 1 : 2 13 25

G 53 14 1 : 1 10 24

H 68 8 1 : 1 10 18

I 75 13 4 : 2 24 37

J 81 5 2 : 2 10 15

K 86 5 4 : 2 11 16

L 93 14 2 : 3 22 36

M 106 3 2 : 2 7 10

N 107 15 4 : 3 14 29

Average wait : 151/14 = 10.786

Average response : 23/14 = 1.643

Average turnaround : 287/14 = 20.500

Throughput: 14/136 = 0.102941 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAABABABCBCBCCCCCCCCCDCDCDDDEDFEDFEDFEDFFFFGFGFGFGFGFGGGGGHGHGHGHGIHIHIHJIHJIKJIKJIKJLIKLIKLILILIMLNIMLNMLNLNLNLNLNLNLNNNNNNN

Test no: 1

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 3 : 0 0 16

B 17 3 4 : 0 0 3

C 22 8 2 : 0 1 9

D 28 11 4 : 1 8 19

E 36 9 3 : 1 12 21

F 39 16 4 : 2 26 42

G 55 5 2 : 2 7 12

H 62 12 2 : 2 28 40

I 68 13 3 : 2 42 55

J 82 9 2 : 2 36 45

K 83 15 1 : 3 42 57

L 90 10 4 : 4 36 46

M 98 5 4 : 5 25 30

N 100 5 2 : 6 25 30

Average wait : 288/14 = 20.571

Average response : 30/14 = 2.143

Average turnaround : 425/14 = 30.357

Throughput: 14/140 = 0.100000 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAA.BBB..CCCCCCCDCDDDDDDEDEDFEDFEDFEFEFEFEFEGFGFGFGHFGHFHIFHIFHIFHIFHIHJIKHJIKHJILKHJILKHJMILNKJMILNKJMILNKJMILNKJMLNKLKLKLKKKK

All 2 tests for [Round Robin] done, averages:

Wait : 15.679

Response : 1.893

Turnaround: 25.429

Throughput: 10.147 per 100 quanta

\*\*\* Testing algorithm: Highest Priority (non-preemptive) \*\*\*

Test no: 0

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 2 : 0 0 16

B 13 5 4 : 3 3 8

C 17 13 4 : 4 4 17

D 31 9 3 : 3 3 12

E 38 4 4 : 39 39 43

F 39 12 1 : 4 4 16

G 53 14 1 : 2 2 16

H 68 8 1 : 1 1 9

I 75 13 4 : 11 11 24

J 81 5 2 : 0 0 5

K 86 5 4 : ~ ~ ~

L 93 14 2 : 6 6 20

M 106 3 2 : ~ ~ ~

N 107 15 4 : ~ ~ ~

Average wait : 73/11 = 6.636

Average response : 73/11 = 6.636

Average turnaround : 186/11 = 16.909

Throughput: 11/113 = 0.097345 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAABBBBBCCCCCCCCCCCCCDDDDDDDDDFFFFFFFFFFFFGGGGGGGGGGGGGGHHHHHHHHEEEEJJJJJIIIIIIIIIIIIILLLLLLLLLLLLLL

Test no: 1

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 3 : 0 0 16

B 17 3 4 : 0 0 3

C 22 8 2 : 0 0 8

D 28 11 4 : 2 2 13

E 36 9 3 : 5 5 14

F 39 16 4 : 11 11 27

G 55 5 2 : 11 11 16

H 62 12 2 : 9 9 21

I 68 13 3 : ~ ~ ~

J 82 9 2 : 16 16 25

K 83 15 1 : 0 0 15

L 90 10 4 : ~ ~ ~

M 98 5 4 : ~ ~ ~

N 100 5 2 : ~ ~ ~

Average wait : 54/10 = 5.400

Average response : 54/10 = 5.400

Average turnaround : 158/10 = 15.800

Throughput: 10/107 = 0.093458 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAA.BBB..CCCCCCCCDDDDDDDDDDDEEEEEEEEEFFFFFFFFFFFFFFFFGGGGGHHHHHHHHHHHHKKKKKKKKKKKKKKKJJJJJJJJJ

All 2 tests for [Highest Priority (non-preemptive)] done, averages:

Wait : 6.018

Response : 6.018

Turnaround: 16.355

Throughput: 9.540 per 100 quanta

\*\*\* Testing algorithm: Highest Priority (preemptive) \*\*\*

Test no: 0

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 2 : 0 0 16

B 13 5 4 : 3 3 8

C 17 13 4 : 4 51 64

D 31 9 3 : 0 12 21

E 38 4 4 : 29 37 41

F 39 12 1 : 0 0 12

G 53 14 1 : 0 0 14

H 68 8 1 : 0 0 8

I 75 13 4 : 11 28 41

J 81 5 2 : 0 0 5

K 86 5 4 : ~ ~ ~

L 93 14 2 : 0 0 14

M 106 3 2 : 1 1 4

N 107 15 4 : ~ ~ ~

Average wait : 132/12 = 11.000

Average response : 48/12 = 4.000

Average turnaround : 248/12 = 20.667

Throughput: 12/116 = 0.103448 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAABBBBBCCCCCCCCCCDDDDDDDDFFFFFFFFFFFFDCGGGGGGGGGGGGGGEHHHHHHHHEEECCJJJJJIIIIIIILLLLLLLLLLLLLLMMMIIIIII

Test no: 1

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 3 : 0 0 16

B 17 3 4 : 0 0 3

C 22 8 2 : 0 0 8

D 28 11 4 : 2 11 22

E 36 9 3 : 0 0 9

F 39 16 4 : 11 65 81

G 55 5 2 : 0 0 5

H 62 12 2 : 0 0 12

I 68 13 3 : 6 30 43

J 82 9 2 : 0 15 24

K 83 15 1 : 0 0 15

L 90 10 4 : ~ ~ ~

M 98 5 4 : ~ ~ ~

N 100 5 2 : ~ ~ ~

Average wait : 121/11 = 11.000

Average response : 19/11 = 1.727

Average turnaround : 238/11 = 21.636

Throughput: 11/120 = 0.091667 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAA.BBB..CCCCCCCCDDDDDDEEEEEEEEEDDDDDFFFFFGGGGGFFHHHHHHHHHHHHIIIIIIIIJKKKKKKKKKKKKKKKJJJJJJJJIIIIIFFFFFFFFF

All 2 tests for [Highest Priority (preemptive)] done, averages:

Wait : 11.000

Response : 2.864

Turnaround: 21.152

Throughput: 9.756 per 100 quanta

\*\*\* Testing algorithm: HPF-Aging (non-preemptive) \*\*\*

Test no: 0

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 2 : 0 0 16

B 13 5 4 : 3 3 8

C 17 13 4 : 4 4 17

D 31 9 3 : 3 3 12

E 38 4 4 : 17 17 21

F 39 12 1 : 4 4 16

G 53 14 1 : 6 6 20

H 68 8 1 : 5 5 13

I 75 13 4 : 11 11 24

J 81 5 2 : 0 0 5

K 86 5 4 : ~ ~ ~

L 93 14 2 : 6 6 20

M 106 3 2 : ~ ~ ~

N 107 15 4 : ~ ~ ~

Average wait : 59/11 = 5.364

Average response : 59/11 = 5.364

Average turnaround : 172/11 = 15.636

Throughput: 11/113 = 0.097345 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAABBBBBCCCCCCCCCCCCCDDDDDDDDDFFFFFFFFFFFFEEEEGGGGGGGGGGGGGGHHHHHHHHJJJJJIIIIIIIIIIIIILLLLLLLLLLLLLL

Test no: 1

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 3 : 0 0 16

B 17 3 4 : 0 0 3

C 22 8 2 : 0 0 8

D 28 11 4 : 2 2 13

E 36 9 3 : 5 5 14

F 39 16 4 : 11 11 27

G 55 5 2 : 11 11 16

H 62 12 2 : 9 9 21

I 68 13 3 : 15 15 28

J 82 9 2 : ~ ~ ~

K 83 15 1 : 13 13 28

L 90 10 4 : ~ ~ ~

M 98 5 4 : ~ ~ ~

N 100 5 2 : ~ ~ ~

Average wait : 66/10 = 6.600

Average response : 66/10 = 6.600

Average turnaround : 174/10 = 17.400

Throughput: 10/111 = 0.090090 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAA.BBB..CCCCCCCCDDDDDDDDDDDEEEEEEEEEFFFFFFFFFFFFFFFFGGGGGHHHHHHHHHHHHIIIIIIIIIIIIIKKKKKKKKKKKKKKK

All 2 tests for [HPF-Aging (non-preemptive)] done, averages:

Wait : 5.982

Response : 5.982

Turnaround: 16.518

Throughput: 9.372 per 100 quanta

\*\*\* Testing algorithm: HPF-Aging (preemptive) \*\*\*

Test no: 0

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 2 : 0 0 16

B 13 5 4 : 3 3 8

C 17 13 4 : 4 4 17

D 31 9 3 : 3 15 24

E 38 4 4 : 17 17 21

F 39 12 1 : 0 0 12

G 53 14 1 : 6 6 20

H 68 8 1 : 5 5 13

I 75 13 4 : 11 11 24

J 81 5 2 : 0 0 5

K 86 5 4 : ~ ~ ~

L 93 14 2 : 6 6 20

M 106 3 2 : ~ ~ ~

N 107 15 4 : ~ ~ ~

Average wait : 67/11 = 6.091

Average response : 55/11 = 5.000

Average turnaround : 180/11 = 16.364

Throughput: 11/113 = 0.097345 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAABBBBBCCCCCCCCCCCCCDDDDDFFFFFFFFFFFFDDDDEEEEGGGGGGGGGGGGGGHHHHHHHHJJJJJIIIIIIIIIIIIILLLLLLLLLLLLLL

Test no: 1

ID Arrival Burst Priority : Response Wait Turnaround

A 0 16 3 : 0 0 16

B 17 3 4 : 0 0 3

C 22 8 2 : 0 0 8

D 28 11 4 : 2 2 13

E 36 9 3 : 5 5 14

F 39 16 4 : 11 11 27

G 55 5 2 : 11 11 16

H 62 12 2 : 9 9 21

I 68 13 3 : 15 15 28

J 82 9 2 : ~ ~ ~

K 83 15 1 : 13 13 28

L 90 10 4 : ~ ~ ~

M 98 5 4 : ~ ~ ~

N 100 5 2 : ~ ~ ~

Average wait : 66/10 = 6.600

Average response : 66/10 = 6.600

Average turnaround : 174/10 = 17.400

Throughput: 10/111 = 0.090090 per single quantum

Execution chart:

0 1 2 3 4 5 6 7 8 9 10 11 12

0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8 0 2 4 6 8

AAAAAAAAAAAAAAAA.BBB..CCCCCCCCDDDDDDDDDDDEEEEEEEEEFFFFFFFFFFFFFFFFGGGGGHHHHHHHHHHHHIIIIIIIIIIIIIKKKKKKKKKKKKKKK

All 2 tests for [HPF-Aging (preemptive)] done, averages:

Wait : 6.345

Response : 5.800

Turnaround: 16.882

Throughput: 9.372 per 100 quanta

C:\Users\jw\Desktop\spring18\opsys\os\hw2>