**University of Michigan – Dearborn**

**CIS 200 – Computer Science 2**

**Project 04**

Nahrin Sharna

[nsharna@umich.edu](mailto:nsharna@umich.edu)

04/17/2019

# Source Code:

## jobStack.h

#include <iostream>

struct job {

char jobType;

int processingTime;

int typeNumber;

int jobNumber;

int arrivalTime;

int waitTime = 0;

};

class jobStack

{

private:

job jobSet[5000];

int jobCount = 0;

public:

jobStack();

void push(job inputJob);

job pop();

bool isEmpty();

bool isFull();

void print();

int peekArrivalTime();

void quickSort(int low, int high);

void swap(job\* a, job\* b);

int partition(int low, int high);

int getCount() { return jobCount; }

~jobStack();

};

## jobStack.cpp

//jobStack.cpp

#include "jobStack.h"

jobStack::jobStack()

{

}

//This function adds a new job to the top of the stack\*/

void jobStack::push(job inputJob)

{

jobCount++;

jobSet[jobCount - 1] = inputJob;

}

//Removes the object on the top of the stack and returns it\*/

job jobStack::pop()

{

job tempJob;

tempJob = jobSet[jobCount - 1];

jobSet[jobCount - 1] = { ' ',0, 0,0,0 };

jobCount--;

return tempJob;

}

//Returns true if there are no objects in the stack\*/

bool jobStack::isEmpty()

{

if (jobCount == 0) {

return true;

}

else {

return false;

}

}

//Returns true if there are 5000 objects in the stack\*/

bool jobStack::isFull()

{

if (jobCount == 5000) {

return true;

}

else {

return false;

}

}

//Prints out the objects in the stack\*/

void jobStack::print()

{

for (int arr\_i = jobCount - 1; arr\_i >= 0; arr\_i--) {

std::cout << "Type " << jobSet[arr\_i].jobType << " "

<< "Arrival Time " << jobSet[arr\_i].arrivalTime << " "

<< "Job Number " << jobSet[arr\_i].jobNumber << " "

<< "Type Number " << jobSet[arr\_i].typeNumber << " "

<< "Processing Time " << jobSet[arr\_i].processingTime << std::endl;

}

}

//Returns the arrival time of the top job object in the stack\*/

int jobStack::peekArrivalTime()

{

return jobSet[jobCount - 1].arrivalTime;

}

// The main function that implements QuickSort

void jobStack::quickSort(int low, int high)

{

if (low < high)

{

int pi = partition(low, high);

quickSort(low, pi - 1);

quickSort(pi + 1, high);

}

}

// A utility function to swap two elements

void jobStack::swap(job \* a, job \* b)

{

job t = \*a;

\*a = \*b;

\*b = t;

}

int jobStack::partition(int low, int high)

{

job pivot = jobSet[high];

int i = (low - 1);

for (int j = low; j <= high - 1; j++)

{

if (jobSet[j].arrivalTime >= pivot.arrivalTime)

{

i++;

swap(&jobSet[i], &jobSet[j]);

}

}

swap(&jobSet[i + 1], &jobSet[high]);

return (i + 1);

}

jobStack::~jobStack()

{

}

## Processor.h

#pragma once

#include "jobStack.h"

class Processor

{

private:

job currentJob;

int jobCount = 0;

public:

Processor();

void push(job inputJob);

job pop();

bool isEmpty();

bool isFull();

bool isComplete();

void processJobOne();

job peekJob() { return currentJob; }

~Processor();

};

## Processor.cpp

#include "Processor.h"

Processor::Processor()

{

}

//Adds an item to the stack

void Processor::push(job inputJob)

{

if (jobCount == 0) {

currentJob = inputJob;

jobCount++;

}

}

//Removes an item from the stack

job Processor::pop()

{

job tempJob = currentJob;

currentJob = {};

jobCount--;

return tempJob;

}

//Returns true when the stack is empty

bool Processor::isEmpty()

{

if (jobCount == 0) {

return true;

}

else {

return false;

}

}

//Returns true when the stack is full(max of 1)

bool Processor::isFull()

{

if (jobCount == 1) {

return true;

}

else {

return false;

}

}

//Returns true when job in the processor has 0 time remaining

bool Processor::isComplete()

{

if (currentJob.processingTime == 0) {

return true;

}

else {

return false;

}

}

//Decreases the processing time in an active job in a processor

void Processor::processJobOne() {

currentJob.processingTime--;

}

Processor::~Processor()

{

}

## minHeap.h

//minHeap.h

#pragma once

#include <iostream>

#include "jobStack.h"

// A class for Min Heap

class MinHeap

{

job \*heaparray;

int capacity;

int heap\_size;

public:

// Constructor

MinHeap(int capacity);

// to heapify a subtree with root at given index

void MinHeapify(int);

// to get index of parent of node at index i

int parent(int i) { return (i - 1) / 2; }

// to get index of left child of node at index i

int left(int i) { return (2 \* i + 1); }

// to get index of right child of node at index i

int right(int i) { return (2 \* i + 2); }

// to extract the root which is the minimum element

job extractMin();

// Returns the minimum key (key at root) from min heap

job getMin() { return heaparray[0]; }

// Inserts a new key 'k'

void insertKey(job k);

bool isEmpty();

void swap(job \*x, job \*y);

void print();

int size() { return heap\_size; }

void addWaitTime();

int getRemainingWaitTime();

};

## minHeap.cpp

//minHeap.cpp

#include "minHeap.h"

// Constructor: Builds a heap from a given array a[] of given size

MinHeap::MinHeap(int cap)

{

heap\_size = 0;

capacity = cap;

heaparray = new job[cap];

}

// Inserts a new key 'k'

void MinHeap::insertKey(job k)

{

if (heap\_size == capacity)

{

std::cout << "\nOverflow: Could not insertKey\n";

return;

}

// First insert the new key at the end

heap\_size++;

int i = heap\_size - 1;

heaparray[i] = k;

// Fix the min heap property if it is violated

while (i != 0 && (heaparray[parent(i)].processingTime > heaparray[i].processingTime || heaparray[i].jobType == 'D'))

{

swap(&heaparray[i], &heaparray[parent(i)]);

i = parent(i);

}

//fixes problem where root isn't swapped out

if (i == 0) {

if (heap\_size > 1 && (heaparray[1].processingTime < heaparray[0].processingTime || heaparray[1].jobType == 'D')) {

swap(&heaparray[1], &heaparray[0]);

}

if (heap\_size > 2 && (heaparray[2].processingTime < heaparray[0].processingTime || heaparray[2].jobType == 'D')) {

swap(&heaparray[2], &heaparray[0]);

}

}

}

//Returns true if there are no objects in the heap

bool MinHeap::isEmpty()

{

if (heap\_size == 0) {

return true;

}

else {

return false;

}

}

// A utility function to swap two elements

void MinHeap::swap(job \* x, job \* y)

{

job temp = \*x;

\*x = \*y;

\*y = temp;

}

//Prints out the objects in the Heap, starting at the root

void MinHeap::print()

{

for (int arr\_i = 0; arr\_i < heap\_size; arr\_i++) {

std::cout << "Type " << heaparray[arr\_i].jobType << " "

<< "Arrival Time " << heaparray[arr\_i].arrivalTime << " "

<< "Job Number " << heaparray[arr\_i].jobNumber << " "

<< "Type Number " << heaparray[arr\_i].typeNumber << " "

<< "Processing Time " << heaparray[arr\_i].processingTime << std::endl;

}

}

//Increases the wait time in all job objects in the queue

void MinHeap::addWaitTime()

{

if (heap\_size > 0) {

for (int arr\_i = 0; arr\_i < heap\_size; arr\_i++) {

heaparray[arr\_i].waitTime++;

}

}

}

//Returns the sum of the wait times of all job objects

int MinHeap::getRemainingWaitTime()

{

int totalWait = 0;

if (heap\_size > 0) {

for (int arr\_i = 0; arr\_i < heap\_size; arr\_i++) {

totalWait = totalWait + heaparray[arr\_i].waitTime;

}

}

return totalWait;

}

// Method to remove minimum element (or root) from min heap

job MinHeap::extractMin()

{

if (heap\_size <= 0) {

return { ' ',0,0,0,0 };

}

if (heap\_size == 1)

{

heap\_size--;

return heaparray[0];

}

// Store the minimum value, and remove it from heap

job root = heaparray[0];

heaparray[0] = heaparray[heap\_size - 1];

heap\_size--;

MinHeapify(0);

return root;

}

// A recursive method to heapify a subtree with root at given index

void MinHeap::MinHeapify(int i)

{

int l = left(i);

int r = right(i);

int smallest = i;

if (l < heap\_size && heaparray[l].processingTime < heaparray[i].processingTime)

smallest = l;

if (r < heap\_size && heaparray[r].processingTime < heaparray[smallest].processingTime)

smallest = r;

if (smallest != i)

{

swap(&heaparray[i], &heaparray[smallest]);

MinHeapify(smallest);

}

}

## Source.cpp (main)

/\*

Author: Nahrin Sharna

Creation Date: 04/06/2019

Modification Dtae: 04/16/2019

Purpose: The program will simulate a processor using a priority queue.

\*/

#include "jobStack.h"

#include "minHeap.h"

#include <random>

#include "Processor.h"

#include <fstream>

#include <iostream>

using namespace std;

void createJobStack(jobStack &);

int main()

{

jobStack inputJobs;

MinHeap jobHeap(5000);

ofstream logFile;

logFile.open("log.txt");

float heapAvg = 0;

int maxQueue = 0;

long int idleCount = 0;

int jobsCompleted = 0;

int aCount = 0, bCount = 0, cCount = 0, dCount = 0;

int activeCycle = 0, totalProcessingTime = 0;

int jobsInterrupted = 0;

int totalWaitTime = 0;

float averageWaitTime = 0;

int userCPU;

createJobStack(inputJobs);

//inputJobs.print();

cout << "Welcome to the Test Processor Program!" << endl;

do {

cout << "How many processors would you like to use in this test?" << endl;

cin >> userCPU;

if (userCPU < 0) {

cout << "Invalid Input" << endl;

}

} while (userCPU < 1);

Processor \*CPU = new Processor[userCPU];

//main program loop

for (int time = 0; time < 10000; time++) {

if (time >= 500) {

logFile << time << ") ";

}

//complete job

for (int i\_cpu = 0; i\_cpu < userCPU; i\_cpu++) {

if (CPU[i\_cpu].isComplete() && CPU[i\_cpu].isFull()) {

job tempJob;

tempJob = CPU[i\_cpu].pop();

if (time >= 500) {

logFile << "Job " << tempJob.jobType << " " << tempJob.typeNumber << " Completed; ";

jobsCompleted++;

}

}

}

//add to the heap

while (inputJobs.peekArrivalTime() == time) {

job tempJob = inputJobs.pop();

jobHeap.insertKey(tempJob);

if (time >= 500) {

logFile << "Arrival Job " << tempJob.jobType << ": Overall Job " << tempJob.jobNumber

<< ", Job " << tempJob.jobType << " " << tempJob.typeNumber

<< ", Processing Time " << tempJob.processingTime << "; ";

//count job types

switch (tempJob.jobType)

{

case 'A':

aCount++;

break;

case 'B':

bCount++;

break;

case 'C':

cCount++;

break;

case 'D':

dCount++;

break;

}

}

}

//interrupt for type D jobs

for (int i\_cpu = 0; i\_cpu < userCPU; i\_cpu++) {

if (CPU[i\_cpu].isEmpty() || jobHeap.size() == 0) {

continue;

}

job temp = jobHeap.getMin();

if (temp.jobType == 'D') {

job displacedJob;

job Djob;

displacedJob = CPU[i\_cpu].pop();

Djob = jobHeap.extractMin();

CPU[i\_cpu].push(Djob);

jobHeap.insertKey(displacedJob);

if (time >= 500) {

jobsInterrupted++;

logFile << "Interrupt Job " << displacedJob.jobType << " " << displacedJob.typeNumber

<< ", Total Interrupted Jobs: " << jobsInterrupted << ", New high priority job goes into Processor " << i\_cpu + 1

<< ", Job " << displacedJob.jobType << " " << displacedJob.typeNumber

<< "Added to heap with processing time " << displacedJob.processingTime << ";";

}

}

else {

break;

}

}

//Add to CPU

for (int i\_cpu = 0; i\_cpu < userCPU; i\_cpu++) {

if (jobHeap.isEmpty()) {

if (time >= 500) {

logFile << "Heap Empty; ";

}

break;

}

else if (CPU[i\_cpu].isEmpty()) {

job tempJob = jobHeap.extractMin();

totalWaitTime = totalWaitTime + tempJob.waitTime;

tempJob.waitTime = 0; //reset in case it is displaced in interrupt

CPU[i\_cpu].push(tempJob);

if (time >= 500) {

logFile << "Begin Processing Job " << tempJob.jobType << " " << tempJob.typeNumber

<< " in CPU " << i\_cpu + 1 << " , end time " << time + tempJob.processingTime << "; ";

}

}

}

//Process

bool activeJob = false;

for (int i\_cpu = 0; i\_cpu < userCPU; i\_cpu++) {

if (CPU[i\_cpu].isEmpty()) {

if (time >= 500) {

logFile << "CPU " << i\_cpu + 1 << ": Idle Time ";

idleCount++;

}

}

else {

job tempJob = CPU[i\_cpu].peekJob();

CPU[i\_cpu].processJobOne();

activeJob = true;

if (time >= 500) {

logFile << "CPU " << i\_cpu + 1 << ": Job " << tempJob.jobType << " " << tempJob.jobNumber << "; ";

totalProcessingTime++;

}

}

}

jobHeap.addWaitTime();

if (activeJob == true && time >= 500)

activeCycle++;

if (time >= 500) {

heapAvg = heapAvg + (jobHeap.size() - heapAvg) / time;

if (jobHeap.size() > maxQueue)

maxQueue = jobHeap.size();

logFile << endl;

}

}

//inputJobs.print();

//jobHeap.print();

//final report

int totalJobsArrived = aCount + bCount + cCount + dCount;

totalWaitTime = totalWaitTime + jobHeap.getRemainingWaitTime();

averageWaitTime = totalWaitTime / totalJobsArrived;

logFile << endl << "Performance Metrics - Calculated from cycle 500 on" << endl;

logFile << "Final Queue Size: " << jobHeap.size() << endl;

logFile << "Average queue size: " << heapAvg << endl;

logFile << "Average time in queue: " << averageWaitTime << " time units" << endl;

logFile << "Idle time: " << idleCount << " time units." << endl;

logFile << "Total Jobs Arrived: " << totalJobsArrived << endl;

logFile << "Total number of jobs A arrived: " << aCount << endl;

logFile << "Total number of jobs B arrived: " << bCount << endl;

logFile << "Total number of jobs C arrived: " << cCount << endl;

logFile << "Total number of jobs D arrived: " << dCount << endl;

logFile << "Total wait time in queue: " << totalWaitTime << " time units." << endl;

logFile << "Maximum jobs in queue: " << maxQueue << endl;

logFile << "Jobs interrupted " << jobsInterrupted << endl;

logFile << "Total jobs completed: " << jobsCompleted << endl;

logFile << "Number of processor(s) used: " << userCPU << endl;

logFile << "Total number of time units the processors(s) run: " << activeCycle << endl;

logFile << "Total time processor(s) spent processing is: " << totalProcessingTime << " time units" << endl;

cout << endl << "Performance Metrics - Calculated from cycle 500 on" << endl;

cout << "Final Queue Size: " << jobHeap.size() << endl;

cout << "Average queue size: " << heapAvg << endl;

cout << "Average time in queue: " << averageWaitTime << " time units" << endl;

cout << "Idle time: " << idleCount << " time units." << endl;

cout << "Total Jobs Arrived: " << totalJobsArrived << endl;

cout << "Total number of jobs A arrived: " << aCount << endl;

cout << "Total number of jobs B arrived: " << bCount << endl;

cout << "Total number of jobs C arrived: " << cCount << endl;

cout << "Total number of jobs D arrived: " << dCount << endl;

cout << "Total wait time in queue: " << totalWaitTime << " time units." << endl;

cout << "Maximum jobs in queue: " << maxQueue << endl;

cout << "Jobs interrupted " << jobsInterrupted << endl;

cout << "Total jobs completed: " << jobsCompleted << endl;

cout << "Number of processor(s) used: " << userCPU << endl;

cout << "Total number of time units the processors(s) run: " << activeCycle << endl;

cout << "Total time processor(s) spent processing is: " << totalProcessingTime << " time units" << endl;

logFile.close();

//jobHeap.print();

cout << "Report information is in the file log.txt" << endl;

cout << "Thank you, have a nice day!" << endl;

system("pause");

return 0;

}

//Description: Creates the job stack

//Precondition : a jobStack object

//Postcondition : returns the job stack

void createJobStack(jobStack &inputJobs)

{

jobStack tempJobs;

int jobCount = 0;

job tempA = { 'A', 0,0,0,0, 0 };

job tempB = { 'B', 0,0,0,0, 0 };

job tempC = { 'C', 0,0,0,0, 0 };

job tempD = { 'D', 0,0,0,0, 0 };

for (int time = 0; time < 10000; time++) {

//create A Type jobs

if (time % 5 == 0) {

tempA.arrivalTime = time + 4 + rand() % 3;

if (tempA.arrivalTime < 10000) {

jobCount++;

tempA.jobNumber = jobCount;

tempA.processingTime = 1 + rand() % 5;

tempA.typeNumber++;

tempJobs.push(tempA);

}

}

//create type B jobs

if (time % 10 == 0) {

tempB.arrivalTime = time + 9 + rand() % 3;

if (tempB.arrivalTime < 10000) {

jobCount++;

tempB.jobNumber = jobCount;

tempB.processingTime = 6 + rand() % 5;

tempB.typeNumber++;

tempJobs.push(tempB);

}

}

//create type C jobs

if (time % 25 == 0) {

tempC.arrivalTime = time + 24 + rand() % 3;

if (tempC.arrivalTime < 10000) {

jobCount++;

tempC.jobNumber = jobCount;

tempC.processingTime = 11 + rand() % 5;

tempC.typeNumber++;

tempJobs.push(tempC);

}

}

//create type D jobs

if (time % 30 == 0) {

tempD.arrivalTime = time + 24 + rand() % 11;

if (tempD.arrivalTime < 10000) {

jobCount++;

tempD.jobNumber = jobCount;

tempD.processingTime = 8 + rand() % 5;

tempD.typeNumber++;

tempJobs.push(tempD);

}

}

}

while (tempJobs.isEmpty() != true) {

inputJobs.push(tempJobs.pop());

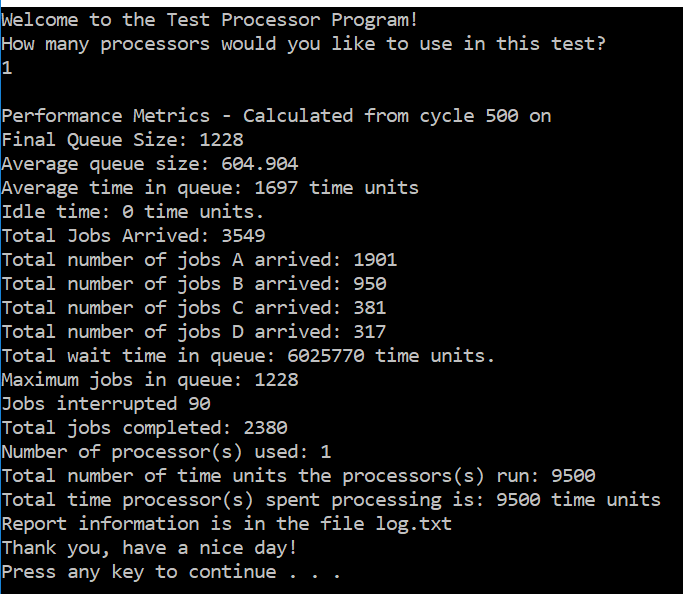
}

inputJobs.quickSort(0, inputJobs.getCount() - 1);

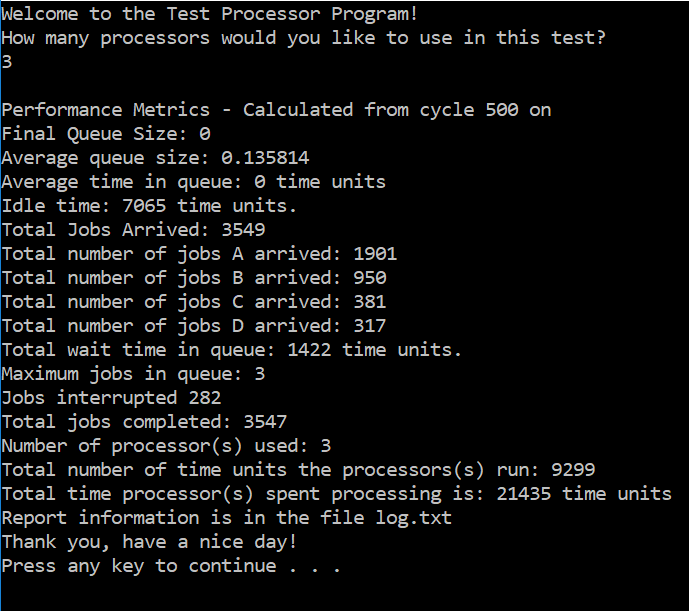
}

# Screenshots:

## Matrix for 1 Processor:



## Matrix for the most effective processor:



# Analysis and determining the efficient processor:

**Explanations:**

For 1 processor, average time is 1697-time units, idle time is 0 and average queue size is 604.904

For 2 processors, average time is 251-time units, idle time is 0 and average queue size is 89.1745

For 3 processors, average time is minimized to 0 while idle time is 7065-time nits and average queue size is .135814

For processor 4, average time is 0 and idle time is 16565-time units.

As it is analyzed, it is observed that using 3 processors will result in minimizing the average time with the lowest value of idle time. Because for 4 processors and so on, the idle time tend to increase. Since, using 3 processors will have least idle time without increasing average time, using 3 processors would be more effective.

# 200 Lines of log file(log.txt):

9800) Arrival Job A: Overall Job 3659, Job A 1960, Processing Time 5; Begin Processing Job A 1960 in CPU 1 , end time 9805; Heap Empty; CPU 1: Job A 3659; CPU 2: Job A 3657; CPU 3: Idle Time

9801) Job A 1959 Completed; Arrival Job C: Overall Job 3652, Job C 392, Processing Time 15; Arrival Job B: Overall Job 3658, Job B 980, Processing Time 8; Begin Processing Job B 980 in CPU 2 , end time 9809; Begin Processing Job C 392 in CPU 3 , end time 9816; CPU 1: Job A 3659; CPU 2: Job B 3658; CPU 3: Job C 3652;

9802) Heap Empty; CPU 1: Job A 3659; CPU 2: Job B 3658; CPU 3: Job C 3652;

9803) Heap Empty; CPU 1: Job A 3659; CPU 2: Job B 3658; CPU 3: Job C 3652;

9804) Arrival Job A: Overall Job 3660, Job A 1961, Processing Time 3; CPU 1: Job A 3659; CPU 2: Job B 3658; CPU 3: Job C 3652;

9805) Job A 1960 Completed; Begin Processing Job A 1961 in CPU 1 , end time 9808; Heap Empty; CPU 1: Job A 3660; CPU 2: Job B 3658; CPU 3: Job C 3652;

9806) Heap Empty; CPU 1: Job A 3660; CPU 2: Job B 3658; CPU 3: Job C 3652;

9807) Heap Empty; CPU 1: Job A 3660; CPU 2: Job B 3658; CPU 3: Job C 3652;

9808) Job A 1961 Completed; Arrival Job D: Overall Job 3655, Job D 327, Processing Time 9; Interrupt Job B 980, Total Interrupted Jobs: 277, New high priority job goes into Processor 2, Job B 980Added to heap with processing time 1;Begin Processing Job B 980 in CPU 1 , end time 9809; Heap Empty; CPU 1: Job B 3658; CPU 2: Job D 3655; CPU 3: Job C 3652;

9809) Job B 980 Completed; Heap Empty; CPU 1: Idle Time CPU 2: Job D 3655; CPU 3: Job C 3652;

9810) Arrival Job B: Overall Job 3661, Job B 981, Processing Time 6; Arrival Job A: Overall Job 3663, Job A 1962, Processing Time 1; Begin Processing Job A 1962 in CPU 1 , end time 9811; CPU 1: Job A 3663; CPU 2: Job D 3655; CPU 3: Job C 3652;

9811) Job A 1962 Completed; Begin Processing Job B 981 in CPU 1 , end time 9817; Heap Empty; CPU 1: Job B 3661; CPU 2: Job D 3655; CPU 3: Job C 3652;

9812) Heap Empty; CPU 1: Job B 3661; CPU 2: Job D 3655; CPU 3: Job C 3652;

9813) Heap Empty; CPU 1: Job B 3661; CPU 2: Job D 3655; CPU 3: Job C 3652;

9814) Heap Empty; CPU 1: Job B 3661; CPU 2: Job D 3655; CPU 3: Job C 3652;

9815) Arrival Job A: Overall Job 3664, Job A 1963, Processing Time 4; CPU 1: Job B 3661; CPU 2: Job D 3655; CPU 3: Job C 3652;

9816) Job C 392 Completed; Begin Processing Job A 1963 in CPU 3 , end time 9820; CPU 1: Job B 3661; CPU 2: Job D 3655; CPU 3: Job A 3664;

9817) Job B 981 Completed; Job D 327 Completed; Heap Empty; CPU 1: Idle Time CPU 2: Idle Time CPU 3: Job A 3664;

9818) Heap Empty; CPU 1: Idle Time CPU 2: Idle Time CPU 3: Job A 3664;

9819) Arrival Job A: Overall Job 3667, Job A 1964, Processing Time 5; Begin Processing Job A 1964 in CPU 1 , end time 9824; Heap Empty; CPU 1: Job A 3667; CPU 2: Idle Time CPU 3: Job A 3664;

9820) Job A 1963 Completed; Heap Empty; CPU 1: Job A 3667; CPU 2: Idle Time CPU 3: Idle Time

9821) Arrival Job B: Overall Job 3665, Job B 982, Processing Time 9; Begin Processing Job B 982 in CPU 2 , end time 9830; Heap Empty; CPU 1: Job A 3667; CPU 2: Job B 3665; CPU 3: Idle Time

9822) Heap Empty; CPU 1: Job A 3667; CPU 2: Job B 3665; CPU 3: Idle Time

9823) Heap Empty; CPU 1: Job A 3667; CPU 2: Job B 3665; CPU 3: Idle Time

9824) Job A 1964 Completed; Arrival Job C: Overall Job 3662, Job C 393, Processing Time 15; Begin Processing Job C 393 in CPU 1 , end time 9839; Heap Empty; CPU 1: Job C 3662; CPU 2: Job B 3665; CPU 3: Idle Time

9825) Heap Empty; CPU 1: Job C 3662; CPU 2: Job B 3665; CPU 3: Idle Time

9826) Arrival Job A: Overall Job 3668, Job A 1965, Processing Time 4; Begin Processing Job A 1965 in CPU 3 , end time 9830; CPU 1: Job C 3662; CPU 2: Job B 3665; CPU 3: Job A 3668;

9827) Heap Empty; CPU 1: Job C 3662; CPU 2: Job B 3665; CPU 3: Job A 3668;

9828) Heap Empty; CPU 1: Job C 3662; CPU 2: Job B 3665; CPU 3: Job A 3668;

9829) Heap Empty; CPU 1: Job C 3662; CPU 2: Job B 3665; CPU 3: Job A 3668;

9830) Job B 982 Completed; Job A 1965 Completed; Arrival Job B: Overall Job 3669, Job B 983, Processing Time 10; Begin Processing Job B 983 in CPU 2 , end time 9840; Heap Empty; CPU 1: Job C 3662; CPU 2: Job B 3669; CPU 3: Idle Time

9831) Arrival Job A: Overall Job 3670, Job A 1966, Processing Time 5; Begin Processing Job A 1966 in CPU 3 , end time 9836; CPU 1: Job C 3662; CPU 2: Job B 3669; CPU 3: Job A 3670;

9832) Heap Empty; CPU 1: Job C 3662; CPU 2: Job B 3669; CPU 3: Job A 3670;

9833) Heap Empty; CPU 1: Job C 3662; CPU 2: Job B 3669; CPU 3: Job A 3670;

9834) Heap Empty; CPU 1: Job C 3662; CPU 2: Job B 3669; CPU 3: Job A 3670;

9835) Arrival Job D: Overall Job 3666, Job D 328, Processing Time 12; Arrival Job A: Overall Job 3672, Job A 1967, Processing Time 4; Interrupt Job C 393, Total Interrupted Jobs: 278, New high priority job goes into Processor 1, Job C 393Added to heap with processing time 4;CPU 1: Job D 3666; CPU 2: Job B 3669; CPU 3: Job A 3670;

9836) Job A 1966 Completed; Begin Processing Job A 1967 in CPU 3 , end time 9840; CPU 1: Job D 3666; CPU 2: Job B 3669; CPU 3: Job A 3672;

9837) CPU 1: Job D 3666; CPU 2: Job B 3669; CPU 3: Job A 3672;

9838) CPU 1: Job D 3666; CPU 2: Job B 3669; CPU 3: Job A 3672;

9839) Arrival Job B: Overall Job 3673, Job B 984, Processing Time 10; CPU 1: Job D 3666; CPU 2: Job B 3669; CPU 3: Job A 3672;

9840) Job B 983 Completed; Job A 1967 Completed; Begin Processing Job C 393 in CPU 2 , end time 9844; Begin Processing Job B 984 in CPU 3 , end time 9850; CPU 1: Job D 3666; CPU 2: Job C 3662; CPU 3: Job B 3673;

9841) Arrival Job A: Overall Job 3674, Job A 1968, Processing Time 1; CPU 1: Job D 3666; CPU 2: Job C 3662; CPU 3: Job B 3673;

9842) CPU 1: Job D 3666; CPU 2: Job C 3662; CPU 3: Job B 3673;

9843) CPU 1: Job D 3666; CPU 2: Job C 3662; CPU 3: Job B 3673;

9844) Job C 393 Completed; Begin Processing Job A 1968 in CPU 2 , end time 9845; Heap Empty; CPU 1: Job D 3666; CPU 2: Job A 3674; CPU 3: Job B 3673;

9845) Job A 1968 Completed; Heap Empty; CPU 1: Job D 3666; CPU 2: Idle Time CPU 3: Job B 3673;

9846) Arrival Job A: Overall Job 3675, Job A 1969, Processing Time 1; Begin Processing Job A 1969 in CPU 2 , end time 9847; Heap Empty; CPU 1: Job D 3666; CPU 2: Job A 3675; CPU 3: Job B 3673;

9847) Job D 328 Completed; Job A 1969 Completed; Heap Empty; CPU 1: Idle Time CPU 2: Idle Time CPU 3: Job B 3673;

9848) Heap Empty; CPU 1: Idle Time CPU 2: Idle Time CPU 3: Job B 3673;

9849) Arrival Job C: Overall Job 3671, Job C 394, Processing Time 11; Begin Processing Job C 394 in CPU 1 , end time 9860; Heap Empty; CPU 1: Job C 3671; CPU 2: Idle Time CPU 3: Job B 3673;

9850) Job B 984 Completed; Arrival Job B: Overall Job 3676, Job B 985, Processing Time 9; Arrival Job A: Overall Job 3678, Job A 1970, Processing Time 1; Begin Processing Job A 1970 in CPU 2 , end time 9851; Begin Processing Job B 985 in CPU 3 , end time 9859; CPU 1: Job C 3671; CPU 2: Job A 3678; CPU 3: Job B 3676;

9851) Job A 1970 Completed; Heap Empty; CPU 1: Job C 3671; CPU 2: Idle Time CPU 3: Job B 3676;

9852) Heap Empty; CPU 1: Job C 3671; CPU 2: Idle Time CPU 3: Job B 3676;

9853) Heap Empty; CPU 1: Job C 3671; CPU 2: Idle Time CPU 3: Job B 3676;

9854) Arrival Job A: Overall Job 3679, Job A 1971, Processing Time 1; Begin Processing Job A 1971 in CPU 2 , end time 9855; Heap Empty; CPU 1: Job C 3671; CPU 2: Job A 3679; CPU 3: Job B 3676;

9855) Job A 1971 Completed; Heap Empty; CPU 1: Job C 3671; CPU 2: Idle Time CPU 3: Job B 3676;

9856) Heap Empty; CPU 1: Job C 3671; CPU 2: Idle Time CPU 3: Job B 3676;

9857) Heap Empty; CPU 1: Job C 3671; CPU 2: Idle Time CPU 3: Job B 3676;

9858) Heap Empty; CPU 1: Job C 3671; CPU 2: Idle Time CPU 3: Job B 3676;

9859) Job B 985 Completed; Arrival Job A: Overall Job 3682, Job A 1972, Processing Time 2; Begin Processing Job A 1972 in CPU 2 , end time 9861; Heap Empty; CPU 1: Job C 3671; CPU 2: Job A 3682; CPU 3: Idle Time

9860) Job C 394 Completed; Arrival Job B: Overall Job 3680, Job B 986, Processing Time 8; Begin Processing Job B 986 in CPU 1 , end time 9868; Heap Empty; CPU 1: Job B 3680; CPU 2: Job A 3682; CPU 3: Idle Time

9861) Job A 1972 Completed; Heap Empty; CPU 1: Job B 3680; CPU 2: Idle Time CPU 3: Idle Time

9862) Heap Empty; CPU 1: Job B 3680; CPU 2: Idle Time CPU 3: Idle Time

9863) Heap Empty; CPU 1: Job B 3680; CPU 2: Idle Time CPU 3: Idle Time

9864) Heap Empty; CPU 1: Job B 3680; CPU 2: Idle Time CPU 3: Idle Time

9865) Heap Empty; CPU 1: Job B 3680; CPU 2: Idle Time CPU 3: Idle Time

9866) Arrival Job A: Overall Job 3683, Job A 1973, Processing Time 2; Begin Processing Job A 1973 in CPU 2 , end time 9868; Heap Empty; CPU 1: Job B 3680; CPU 2: Job A 3683; CPU 3: Idle Time

9867) Heap Empty; CPU 1: Job B 3680; CPU 2: Job A 3683; CPU 3: Idle Time

9868) Job B 986 Completed; Job A 1973 Completed; Arrival Job D: Overall Job 3677, Job D 329, Processing Time 10; Begin Processing Job D 329 in CPU 1 , end time 9878; Heap Empty; CPU 1: Job D 3677; CPU 2: Idle Time CPU 3: Idle Time

9869) Heap Empty; CPU 1: Job D 3677; CPU 2: Idle Time CPU 3: Idle Time

9870) Arrival Job A: Overall Job 3685, Job A 1974, Processing Time 5; Arrival Job B: Overall Job 3684, Job B 987, Processing Time 7; Begin Processing Job A 1974 in CPU 2 , end time 9875; Begin Processing Job B 987 in CPU 3 , end time 9877; CPU 1: Job D 3677; CPU 2: Job A 3685; CPU 3: Job B 3684;

9871) Heap Empty; CPU 1: Job D 3677; CPU 2: Job A 3685; CPU 3: Job B 3684;

9872) Heap Empty; CPU 1: Job D 3677; CPU 2: Job A 3685; CPU 3: Job B 3684;

9873) Heap Empty; CPU 1: Job D 3677; CPU 2: Job A 3685; CPU 3: Job B 3684;

9874) Heap Empty; CPU 1: Job D 3677; CPU 2: Job A 3685; CPU 3: Job B 3684;

9875) Job A 1974 Completed; Arrival Job C: Overall Job 3681, Job C 395, Processing Time 11; Begin Processing Job C 395 in CPU 2 , end time 9886; Heap Empty; CPU 1: Job D 3677; CPU 2: Job C 3681; CPU 3: Job B 3684;

9876) Arrival Job A: Overall Job 3686, Job A 1975, Processing Time 1; CPU 1: Job D 3677; CPU 2: Job C 3681; CPU 3: Job B 3684;

9877) Job B 987 Completed; Begin Processing Job A 1975 in CPU 3 , end time 9878; CPU 1: Job D 3677; CPU 2: Job C 3681; CPU 3: Job A 3686;

9878) Job D 329 Completed; Job A 1975 Completed; Heap Empty; CPU 1: Idle Time CPU 2: Job C 3681; CPU 3: Idle Time

9879) Heap Empty; CPU 1: Idle Time CPU 2: Job C 3681; CPU 3: Idle Time

9880) Arrival Job B: Overall Job 3687, Job B 988, Processing Time 9; Begin Processing Job B 988 in CPU 1 , end time 9889; Heap Empty; CPU 1: Job B 3687; CPU 2: Job C 3681; CPU 3: Idle Time

9881) Arrival Job A: Overall Job 3689, Job A 1976, Processing Time 3; Begin Processing Job A 1976 in CPU 3 , end time 9884; CPU 1: Job B 3687; CPU 2: Job C 3681; CPU 3: Job A 3689;

9882) Heap Empty; CPU 1: Job B 3687; CPU 2: Job C 3681; CPU 3: Job A 3689;

9883) Heap Empty; CPU 1: Job B 3687; CPU 2: Job C 3681; CPU 3: Job A 3689;

9884) Job A 1976 Completed; Arrival Job A: Overall Job 3691, Job A 1977, Processing Time 4; Begin Processing Job A 1977 in CPU 3 , end time 9888; CPU 1: Job B 3687; CPU 2: Job C 3681; CPU 3: Job A 3691;

9885) Heap Empty; CPU 1: Job B 3687; CPU 2: Job C 3681; CPU 3: Job A 3691;

9886) Job C 395 Completed; Heap Empty; CPU 1: Job B 3687; CPU 2: Idle Time CPU 3: Job A 3691;

9887) Heap Empty; CPU 1: Job B 3687; CPU 2: Idle Time CPU 3: Job A 3691;

9888) Job A 1977 Completed; Heap Empty; CPU 1: Job B 3687; CPU 2: Idle Time CPU 3: Idle Time

9889) Job B 988 Completed; Arrival Job B: Overall Job 3692, Job B 989, Processing Time 10; Begin Processing Job B 989 in CPU 1 , end time 9899; Heap Empty; CPU 1: Job B 3692; CPU 2: Idle Time CPU 3: Idle Time

9890) Arrival Job A: Overall Job 3693, Job A 1978, Processing Time 2; Begin Processing Job A 1978 in CPU 2 , end time 9892; Heap Empty; CPU 1: Job B 3692; CPU 2: Job A 3693; CPU 3: Idle Time

9891) Heap Empty; CPU 1: Job B 3692; CPU 2: Job A 3693; CPU 3: Idle Time

9892) Job A 1978 Completed; Heap Empty; CPU 1: Job B 3692; CPU 2: Idle Time CPU 3: Idle Time

9893) Heap Empty; CPU 1: Job B 3692; CPU 2: Idle Time CPU 3: Idle Time

9894) Arrival Job A: Overall Job 3694, Job A 1979, Processing Time 2; Begin Processing Job A 1979 in CPU 2 , end time 9896; Heap Empty; CPU 1: Job B 3692; CPU 2: Job A 3694; CPU 3: Idle Time

9895) Heap Empty; CPU 1: Job B 3692; CPU 2: Job A 3694; CPU 3: Idle Time

9896) Job A 1979 Completed; Heap Empty; CPU 1: Job B 3692; CPU 2: Idle Time CPU 3: Idle Time

9897) Heap Empty; CPU 1: Job B 3692; CPU 2: Idle Time CPU 3: Idle Time

9898) Heap Empty; CPU 1: Job B 3692; CPU 2: Idle Time CPU 3: Idle Time

9899) Job B 989 Completed; Arrival Job A: Overall Job 3696, Job A 1980, Processing Time 3; Begin Processing Job A 1980 in CPU 1 , end time 9902; Heap Empty; CPU 1: Job A 3696; CPU 2: Idle Time CPU 3: Idle Time

9900) Heap Empty; CPU 1: Job A 3696; CPU 2: Idle Time CPU 3: Idle Time

9901) Arrival Job C: Overall Job 3690, Job C 396, Processing Time 12; Arrival Job B: Overall Job 3695, Job B 990, Processing Time 6; Begin Processing Job B 990 in CPU 2 , end time 9907; Begin Processing Job C 396 in CPU 3 , end time 9913; CPU 1: Job A 3696; CPU 2: Job B 3695; CPU 3: Job C 3690;

9902) Job A 1980 Completed; Heap Empty; CPU 1: Idle Time CPU 2: Job B 3695; CPU 3: Job C 3690;

9903) Arrival Job D: Overall Job 3688, Job D 330, Processing Time 11; Interrupt Job B 990, Total Interrupted Jobs: 279, New high priority job goes into Processor 2, Job B 990Added to heap with processing time 4;Begin Processing Job B 990 in CPU 1 , end time 9907; Heap Empty; CPU 1: Job B 3695; CPU 2: Job D 3688; CPU 3: Job C 3690;

9904) Heap Empty; CPU 1: Job B 3695; CPU 2: Job D 3688; CPU 3: Job C 3690;

9905) Arrival Job A: Overall Job 3697, Job A 1981, Processing Time 4; CPU 1: Job B 3695; CPU 2: Job D 3688; CPU 3: Job C 3690;

9906) CPU 1: Job B 3695; CPU 2: Job D 3688; CPU 3: Job C 3690;

9907) Job B 990 Completed; Begin Processing Job A 1981 in CPU 1 , end time 9911; Heap Empty; CPU 1: Job A 3697; CPU 2: Job D 3688; CPU 3: Job C 3690;

9908) Heap Empty; CPU 1: Job A 3697; CPU 2: Job D 3688; CPU 3: Job C 3690;

9909) Heap Empty; CPU 1: Job A 3697; CPU 2: Job D 3688; CPU 3: Job C 3690;

9910) Arrival Job A: Overall Job 3701, Job A 1982, Processing Time 1; CPU 1: Job A 3697; CPU 2: Job D 3688; CPU 3: Job C 3690;

9911) Job A 1981 Completed; Arrival Job B: Overall Job 3698, Job B 991, Processing Time 9; Begin Processing Job A 1982 in CPU 1 , end time 9912; CPU 1: Job A 3701; CPU 2: Job D 3688; CPU 3: Job C 3690;

9912) Job A 1982 Completed; Begin Processing Job B 991 in CPU 1 , end time 9921; Heap Empty; CPU 1: Job B 3698; CPU 2: Job D 3688; CPU 3: Job C 3690;

9913) Job C 396 Completed; Heap Empty; CPU 1: Job B 3698; CPU 2: Job D 3688; CPU 3: Idle Time

9914) Job D 330 Completed; Heap Empty; CPU 1: Job B 3698; CPU 2: Idle Time CPU 3: Idle Time

9915) Heap Empty; CPU 1: Job B 3698; CPU 2: Idle Time CPU 3: Idle Time

9916) Arrival Job A: Overall Job 3702, Job A 1983, Processing Time 4; Begin Processing Job A 1983 in CPU 2 , end time 9920; Heap Empty; CPU 1: Job B 3698; CPU 2: Job A 3702; CPU 3: Idle Time

9917) Heap Empty; CPU 1: Job B 3698; CPU 2: Job A 3702; CPU 3: Idle Time

9918) Heap Empty; CPU 1: Job B 3698; CPU 2: Job A 3702; CPU 3: Idle Time

9919) Heap Empty; CPU 1: Job B 3698; CPU 2: Job A 3702; CPU 3: Idle Time

9920) Job A 1983 Completed; Heap Empty; CPU 1: Job B 3698; CPU 2: Idle Time CPU 3: Idle Time

9921) Job B 991 Completed; Arrival Job B: Overall Job 3703, Job B 992, Processing Time 6; Arrival Job A: Overall Job 3704, Job A 1984, Processing Time 5; Begin Processing Job A 1984 in CPU 1 , end time 9926; Begin Processing Job B 992 in CPU 2 , end time 9927; Heap Empty; CPU 1: Job A 3704; CPU 2: Job B 3703; CPU 3: Idle Time

9922) Heap Empty; CPU 1: Job A 3704; CPU 2: Job B 3703; CPU 3: Idle Time

9923) Heap Empty; CPU 1: Job A 3704; CPU 2: Job B 3703; CPU 3: Idle Time

9924) Heap Empty; CPU 1: Job A 3704; CPU 2: Job B 3703; CPU 3: Idle Time

9925) Arrival Job C: Overall Job 3699, Job C 397, Processing Time 13; Begin Processing Job C 397 in CPU 3 , end time 9938; CPU 1: Job A 3704; CPU 2: Job B 3703; CPU 3: Job C 3699;

9926) Job A 1984 Completed; Arrival Job A: Overall Job 3705, Job A 1985, Processing Time 3; Begin Processing Job A 1985 in CPU 1 , end time 9929; Heap Empty; CPU 1: Job A 3705; CPU 2: Job B 3703; CPU 3: Job C 3699;

9927) Job B 992 Completed; Heap Empty; CPU 1: Job A 3705; CPU 2: Idle Time CPU 3: Job C 3699;

9928) Arrival Job D: Overall Job 3700, Job D 331, Processing Time 9; Interrupt Job A 1985, Total Interrupted Jobs: 280, New high priority job goes into Processor 1, Job A 1985Added to heap with processing time 1;Begin Processing Job A 1985 in CPU 2 , end time 9929; Heap Empty; CPU 1: Job D 3700; CPU 2: Job A 3705; CPU 3: Job C 3699;

9929) Job A 1985 Completed; Heap Empty; CPU 1: Job D 3700; CPU 2: Idle Time CPU 3: Job C 3699;

9930) Arrival Job B: Overall Job 3706, Job B 993, Processing Time 8; Begin Processing Job B 993 in CPU 2 , end time 9938; Heap Empty; CPU 1: Job D 3700; CPU 2: Job B 3706; CPU 3: Job C 3699;

9931) Arrival Job A: Overall Job 3707, Job A 1986, Processing Time 5; CPU 1: Job D 3700; CPU 2: Job B 3706; CPU 3: Job C 3699;

9932) CPU 1: Job D 3700; CPU 2: Job B 3706; CPU 3: Job C 3699;

9933) CPU 1: Job D 3700; CPU 2: Job B 3706; CPU 3: Job C 3699;

9934) CPU 1: Job D 3700; CPU 2: Job B 3706; CPU 3: Job C 3699;

9935) Arrival Job A: Overall Job 3709, Job A 1987, Processing Time 5; CPU 1: Job D 3700; CPU 2: Job B 3706; CPU 3: Job C 3699;

9936) CPU 1: Job D 3700; CPU 2: Job B 3706; CPU 3: Job C 3699;

9937) Job D 331 Completed; Begin Processing Job A 1986 in CPU 1 , end time 9942; CPU 1: Job A 3707; CPU 2: Job B 3706; CPU 3: Job C 3699;

9938) Job B 993 Completed; Job C 397 Completed; Begin Processing Job A 1987 in CPU 2 , end time 9943; Heap Empty; CPU 1: Job A 3707; CPU 2: Job A 3709; CPU 3: Idle Time

9939) Heap Empty; CPU 1: Job A 3707; CPU 2: Job A 3709; CPU 3: Idle Time

9940) Heap Empty; CPU 1: Job A 3707; CPU 2: Job A 3709; CPU 3: Idle Time

9941) Arrival Job A: Overall Job 3712, Job A 1988, Processing Time 5; Arrival Job B: Overall Job 3710, Job B 994, Processing Time 7; Begin Processing Job A 1988 in CPU 3 , end time 9946; CPU 1: Job A 3707; CPU 2: Job A 3709; CPU 3: Job A 3712;

9942) Job A 1986 Completed; Begin Processing Job B 994 in CPU 1 , end time 9949; Heap Empty; CPU 1: Job B 3710; CPU 2: Job A 3709; CPU 3: Job A 3712;

9943) Job A 1987 Completed; Heap Empty; CPU 1: Job B 3710; CPU 2: Idle Time CPU 3: Job A 3712;

9944) Heap Empty; CPU 1: Job B 3710; CPU 2: Idle Time CPU 3: Job A 3712;

9945) Arrival Job A: Overall Job 3713, Job A 1989, Processing Time 5; Begin Processing Job A 1989 in CPU 2 , end time 9950; Heap Empty; CPU 1: Job B 3710; CPU 2: Job A 3713; CPU 3: Job A 3712;

9946) Job A 1988 Completed; Heap Empty; CPU 1: Job B 3710; CPU 2: Job A 3713; CPU 3: Idle Time

9947) Heap Empty; CPU 1: Job B 3710; CPU 2: Job A 3713; CPU 3: Idle Time

9948) Heap Empty; CPU 1: Job B 3710; CPU 2: Job A 3713; CPU 3: Idle Time

9949) Job B 994 Completed; Arrival Job A: Overall Job 3715, Job A 1990, Processing Time 1; Begin Processing Job A 1990 in CPU 1 , end time 9950; Heap Empty; CPU 1: Job A 3715; CPU 2: Job A 3713; CPU 3: Idle Time

9950) Job A 1990 Completed; Job A 1989 Completed; Arrival Job C: Overall Job 3708, Job C 398, Processing Time 13; Begin Processing Job C 398 in CPU 1 , end time 9963; Heap Empty; CPU 1: Job C 3708; CPU 2: Idle Time CPU 3: Idle Time

9951) Arrival Job B: Overall Job 3714, Job B 995, Processing Time 8; Begin Processing Job B 995 in CPU 2 , end time 9959; Heap Empty; CPU 1: Job C 3708; CPU 2: Job B 3714; CPU 3: Idle Time

9952) Heap Empty; CPU 1: Job C 3708; CPU 2: Job B 3714; CPU 3: Idle Time

9953) Heap Empty; CPU 1: Job C 3708; CPU 2: Job B 3714; CPU 3: Idle Time

9954) Arrival Job A: Overall Job 3716, Job A 1991, Processing Time 4; Begin Processing Job A 1991 in CPU 3 , end time 9958; CPU 1: Job C 3708; CPU 2: Job B 3714; CPU 3: Job A 3716;

9955) Heap Empty; CPU 1: Job C 3708; CPU 2: Job B 3714; CPU 3: Job A 3716;

9956) Heap Empty; CPU 1: Job C 3708; CPU 2: Job B 3714; CPU 3: Job A 3716;

9957) Heap Empty; CPU 1: Job C 3708; CPU 2: Job B 3714; CPU 3: Job A 3716;

9958) Job A 1991 Completed; Arrival Job D: Overall Job 3711, Job D 332, Processing Time 12; Interrupt Job C 398, Total Interrupted Jobs: 281, New high priority job goes into Processor 1, Job C 398Added to heap with processing time 5;Begin Processing Job C 398 in CPU 3 , end time 9963; CPU 1: Job D 3711; CPU 2: Job B 3714; CPU 3: Job C 3708;

9959) Job B 995 Completed; Heap Empty; CPU 1: Job D 3711; CPU 2: Idle Time CPU 3: Job C 3708;

9960) Arrival Job B: Overall Job 3717, Job B 996, Processing Time 6; Arrival Job A: Overall Job 3719, Job A 1992, Processing Time 4; Begin Processing Job A 1992 in CPU 2 , end time 9964; CPU 1: Job D 3711; CPU 2: Job A 3719; CPU 3: Job C 3708;

9961) CPU 1: Job D 3711; CPU 2: Job A 3719; CPU 3: Job C 3708;

9962) CPU 1: Job D 3711; CPU 2: Job A 3719; CPU 3: Job C 3708;

9963) Job C 398 Completed; Begin Processing Job B 996 in CPU 3 , end time 9969; CPU 1: Job D 3711; CPU 2: Job A 3719; CPU 3: Job B 3717;

9964) Job A 1992 Completed; Heap Empty; CPU 1: Job D 3711; CPU 2: Idle Time CPU 3: Job B 3717;

9965) Heap Empty; CPU 1: Job D 3711; CPU 2: Idle Time CPU 3: Job B 3717;

9966) Arrival Job A: Overall Job 3720, Job A 1993, Processing Time 5; Begin Processing Job A 1993 in CPU 2 , end time 9971; Heap Empty; CPU 1: Job D 3711; CPU 2: Job A 3720; CPU 3: Job B 3717;

9967) Heap Empty; CPU 1: Job D 3711; CPU 2: Job A 3720; CPU 3: Job B 3717;

9968) Heap Empty; CPU 1: Job D 3711; CPU 2: Job A 3720; CPU 3: Job B 3717;

9969) Job B 996 Completed; Arrival Job B: Overall Job 3721, Job B 997, Processing Time 6; Begin Processing Job B 997 in CPU 3 , end time 9975; CPU 1: Job D 3711; CPU 2: Job A 3720; CPU 3: Job B 3721;

9970) Job D 332 Completed; Heap Empty; CPU 1: Idle Time CPU 2: Job A 3720; CPU 3: Job B 3721;

9971) Job A 1993 Completed; Arrival Job A: Overall Job 3723, Job A 1994, Processing Time 5; Begin Processing Job A 1994 in CPU 1 , end time 9976; Heap Empty; CPU 1: Job A 3723; CPU 2: Idle Time CPU 3: Job B 3721;

9972) Heap Empty; CPU 1: Job A 3723; CPU 2: Idle Time CPU 3: Job B 3721;

9973) Heap Empty; CPU 1: Job A 3723; CPU 2: Idle Time CPU 3: Job B 3721;

9974) Arrival Job A: Overall Job 3724, Job A 1995, Processing Time 5; Begin Processing Job A 1995 in CPU 2 , end time 9979; Heap Empty; CPU 1: Job A 3723; CPU 2: Job A 3724; CPU 3: Job B 3721;

9975) Job B 997 Completed; Arrival Job C: Overall Job 3718, Job C 399, Processing Time 14; Begin Processing Job C 399 in CPU 3 , end time 9989; CPU 1: Job A 3723; CPU 2: Job A 3724; CPU 3: Job C 3718;

9976) Job A 1994 Completed; Heap Empty; CPU 1: Idle Time CPU 2: Job A 3724; CPU 3: Job C 3718;

9977) Heap Empty; CPU 1: Idle Time CPU 2: Job A 3724; CPU 3: Job C 3718;

9978) Heap Empty; CPU 1: Idle Time CPU 2: Job A 3724; CPU 3: Job C 3718;

9979) Job A 1995 Completed; Heap Empty; CPU 1: Idle Time CPU 2: Idle Time CPU 3: Job C 3718;

9980) Arrival Job B: Overall Job 3725, Job B 998, Processing Time 6; Begin Processing Job B 998 in CPU 1 , end time 9986; Heap Empty; CPU 1: Job B 3725; CPU 2: Idle Time CPU 3: Job C 3718;

9981) Arrival Job A: Overall Job 3726, Job A 1996, Processing Time 1; Begin Processing Job A 1996 in CPU 2 , end time 9982; Heap Empty; CPU 1: Job B 3725; CPU 2: Job A 3726; CPU 3: Job C 3718;

9982) Job A 1996 Completed; Heap Empty; CPU 1: Job B 3725; CPU 2: Idle Time CPU 3: Job C 3718;

9983) Heap Empty; CPU 1: Job B 3725; CPU 2: Idle Time CPU 3: Job C 3718;

9984) Arrival Job A: Overall Job 3728, Job A 1997, Processing Time 3; Begin Processing Job A 1997 in CPU 2 , end time 9987; Heap Empty; CPU 1: Job B 3725; CPU 2: Job A 3728; CPU 3: Job C 3718;

9985) Heap Empty; CPU 1: Job B 3725; CPU 2: Job A 3728; CPU 3: Job C 3718;

9986) Job B 998 Completed; Arrival Job D: Overall Job 3722, Job D 333, Processing Time 11; Interrupt Job A 1997, Total Interrupted Jobs: 282, New high priority job goes into Processor 2, Job A 1997Added to heap with processing time 1;Begin Processing Job A 1997 in CPU 1 , end time 9987; Heap Empty; CPU 1: Job A 3728; CPU 2: Job D 3722; CPU 3: Job C 3718;

9987) Job A 1997 Completed; Heap Empty; CPU 1: Idle Time CPU 2: Job D 3722; CPU 3: Job C 3718;

9988) Heap Empty; CPU 1: Idle Time CPU 2: Job D 3722; CPU 3: Job C 3718;

9989) Job C 399 Completed; Arrival Job B: Overall Job 3729, Job B 999, Processing Time 6; Begin Processing Job B 999 in CPU 1 , end time 9995; Heap Empty; CPU 1: Job B 3729; CPU 2: Job D 3722; CPU 3: Idle Time

9990) Arrival Job A: Overall Job 3730, Job A 1998, Processing Time 1; Begin Processing Job A 1998 in CPU 3 , end time 9991; CPU 1: Job B 3729; CPU 2: Job D 3722; CPU 3: Job A 3730;

9991) Job A 1998 Completed; Heap Empty; CPU 1: Job B 3729; CPU 2: Job D 3722; CPU 3: Idle Time

9992) Heap Empty; CPU 1: Job B 3729; CPU 2: Job D 3722; CPU 3: Idle Time

9993) Heap Empty; CPU 1: Job B 3729; CPU 2: Job D 3722; CPU 3: Idle Time

9994) Arrival Job A: Overall Job 3731, Job A 1999, Processing Time 5; Begin Processing Job A 1999 in CPU 3 , end time 9999; CPU 1: Job B 3729; CPU 2: Job D 3722; CPU 3: Job A 3731;

9995) Job B 999 Completed; Heap Empty; CPU 1: Idle Time CPU 2: Job D 3722; CPU 3: Job A 3731;

9996) Heap Empty; CPU 1: Idle Time CPU 2: Job D 3722; CPU 3: Job A 3731;

9997) Job D 333 Completed; Heap Empty; CPU 1: Idle Time CPU 2: Idle Time CPU 3: Job A 3731;

9998) Heap Empty; CPU 1: Idle Time CPU 2: Idle Time CPU 3: Job A 3731;

9999) Job A 1999 Completed; Arrival Job C: Overall Job 3727, Job C 400, Processing Time 15; Arrival Job B: Overall Job 3732, Job B 1000, Processing Time 9; Arrival Job A: Overall Job 3733, Job A 2000, Processing Time 4; Begin Processing Job A 2000 in CPU 1 , end time 10003; Begin Processing Job B 1000 in CPU 2 , end time 10008; Begin Processing Job C 400 in CPU 3 , end time 10014; CPU 1: Job A 3733; CPU 2: Job B 3732; CPU 3: Job C 3727;