Saudi Arabia

King Abdul Aziz University

The faculty of computer and information technology

Computer Science Department

Course Project

CPCS 371 Operating System

Fall 20-21

PREPARED BY:

1780367	Aya Kazzaz	GAR
1805645	Hadeel aloufi	GAR
1780293	Salwa Abbara	GAR
1780735	Mona Hafez	GAR
1806494	Aisha Baskran	GAR

Table of Contents

1. Introduction

In this report, we'll talk a little bit about CPU process scheduling, and more Specifically we'll be discussing Round Robin “RR” and Dynamic Round Robin .

So, what is round robin? what is dynamic round robin ? , and we provide a code written in Java.

And also, We will compare the RR and DRR at the end of the report, and finally we will review the results.

1.2 What is round robin ?

Round Robin is a CPU scheduling algorithm where each process is assigned a fixed time slot in a cyclic way, CPU is assigned to the process based on FCFS for a fixed amount of time. This fixed amount of time is called as **time quantum** or **time slice**.

After the time quantum expires, the running process is preempted and sent to the ready queue. Then, the processor is assigned to the next arrived process.

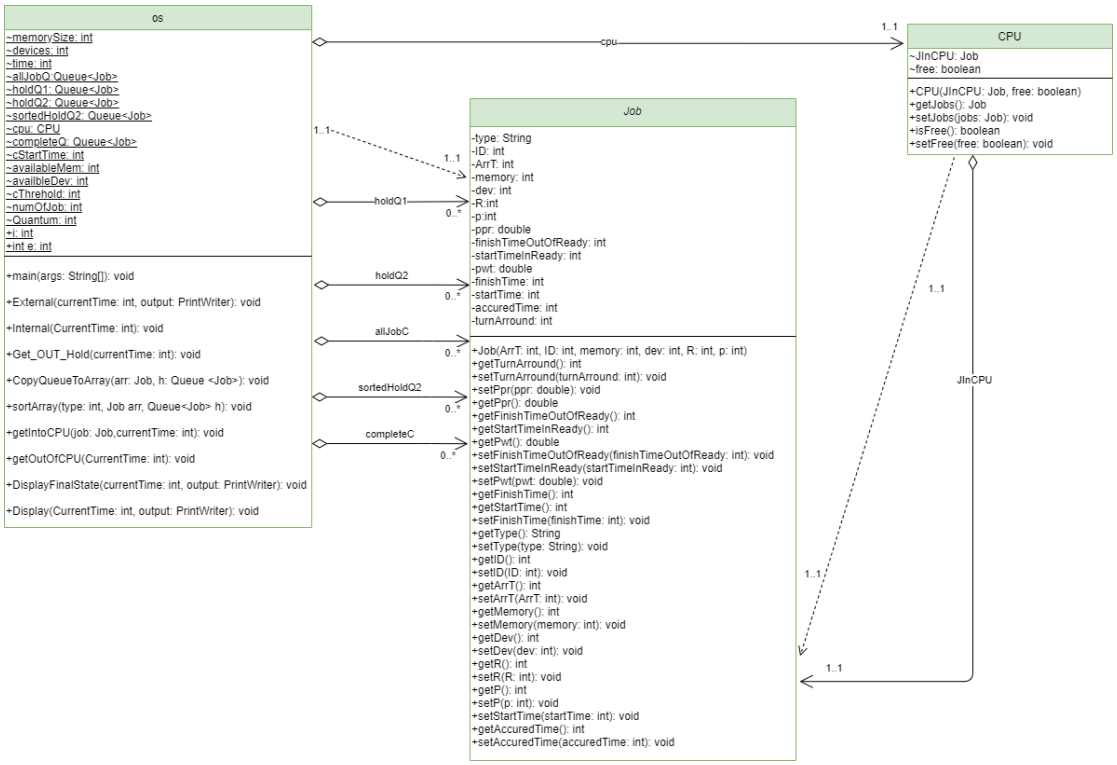
1.3 What is dynamic round robin ?

Reducing time cost in shared OS time is the main goal of researchers interested in CPU scheduling.

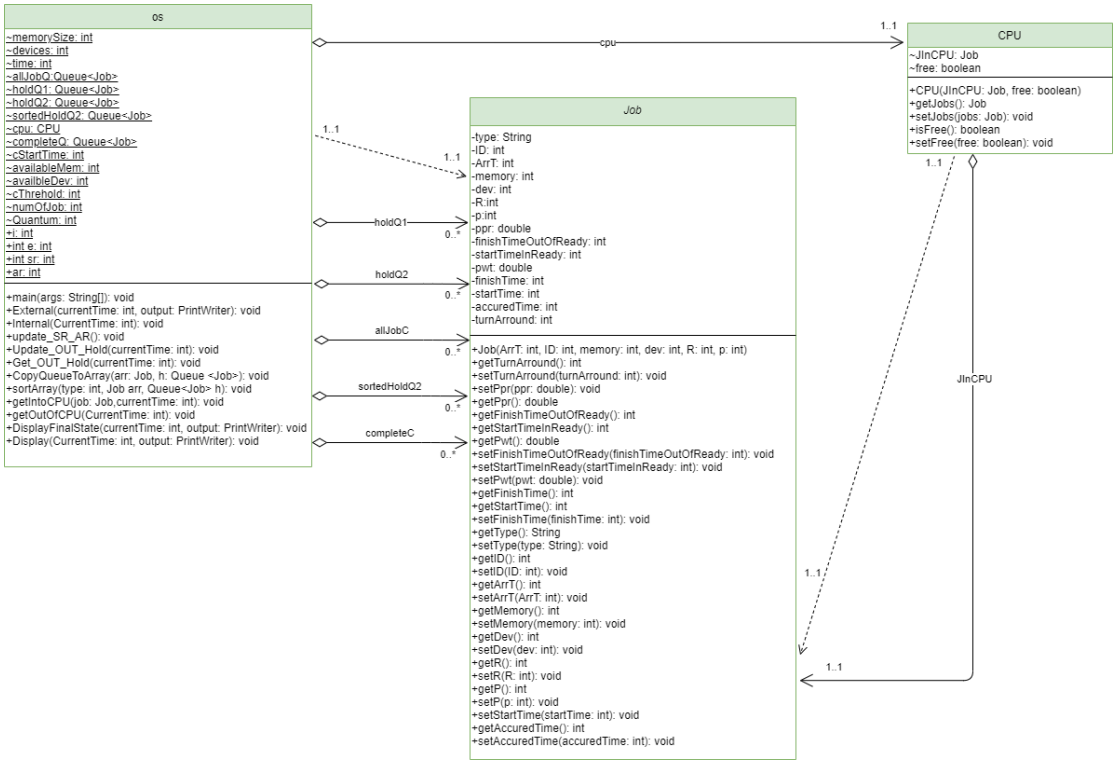
A modified version of the RR algorithm was introduced to combine the advantages of the preferred short process and lower scheduling expense of the RR in order to reduce the average wait time and completion time. Each process in the block is assigned the same time slot based on the batch weight and CPU lifetime.

2. Design based on UML Diagrams

❖ UML static RR



❖ UML Dynamic RR



3. A copy of a printout of your source code

osMain(Dynamic)

```
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
//Aya Kazzaz,Hadeel aloufi,Salwa Abbara,Mona Hafez,Aisha
Baskran,windows10,version1909(OS Build 18363.1256)
//processor:Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 2.00 GHz
//Installed memory(RAM): 8.00 GB(7.90 GB usable)
//compiler name and version:java version "1.8.0_191" ,Java(TM) SE Runtime
Environment

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author WinDows
 */
public class osMain {
 static int memorySize;
static int devices;
static int time;
static Queue<Job>allJobQ=new LinkedList();
static Queue<Job>holdQ1=new LinkedList();
static Queue<Job>holdQ2=new LinkedList();
static Queue<Job>sortedHoldQ2=new LinkedList();
static CPU cpu=new CPU(null,true);
static Queue<Job>completeQ=new LinkedList();
static int cStartTime=0;
static int availableMem=0;
static int availbleDev=0;
static int cThrehold=0;
static int numOfJob=0;
static int Quantum=0;
public static int i;
public static int e;
public static int sr;
```

```
public static int ar;


    public static void main(String[] args) throws FileNotFoundException {
    java.io.File file = new java.io.File("input1.txt");
     // java.io.File file = new java.io.File("input2.txt");you can change between these
3 input files to check outputs files
    //java.io.File file = new java.io.File("input3.txt");
    java.io.File outputFile = new java.io.File("output.txt");
    PrintWriter output=new PrintWriter(outputFile);//print writer to write to the
outputfile

    Scanner input=new Scanner(file);
    while (input.hasNext()) {

     String command1=input.next();

     time=input.nextInt();

     memorySize=Integer.parseInt(input.next().substring(2));
     availableMem=memorySize;

     devices=Integer.parseInt(input.next().substring(2));

       availbleDev=devices;
     Job job;
     numOfJob=0;
     int number = 0;
     do{

         String command2=input.next();

         if(command2.equals("A")){
           int ArrT= Integer.parseInt(input.next());

           int ID=Integer.parseInt(input.next().substring(2));

           int memory=Integer.parseInt(input.next().substring(2));

           int dev=Integer.parseInt(input.next().substring(2));

           int R=Integer.parseInt(input.next().substring(2));

           int p=Integer.parseInt(input.next().substring(2));
```

```
  if(memory<=memorySize&&dev<=devices){
    job=new Job(ArrT,ID,memory,dev,R,p);

    allJobQ.add(job);
    numOfJob++;//counter to count the number of jobs

  }
}else if(command2.equals("D")){

    int time1=input.nextInt();

    if(time1<999999){
     job=new Job(time1,-1,-1,-1,-1,-1);

     allJobQ.add(job);
    }
    else{

    break;
}
}
} while(input.hasNext());
  Job j=allJobQ.poll();

  int currentTime=j.getArrT();

  availableMem-=j.getMemory();
  Quantum=j.getR();//take the first job and set the quantum to its burst time

  getIntoCPU(j,currentTime);

   i = 0;
   e = 0;

   do {

       if (!(allJobQ.isEmpty())) {

           i = allJobQ.peek().getArrT();
       } else {
           i = Integer.MAX_VALUE;
       }


       if (cpu.isFree()==false) {
```

```
            e = cpu.JInCPU.getFinishTime();

        } else {
            e = Integer.MAX_VALUE;
        }
        currentTime = Math.min(i, e);

        if (i < e) {

            External(currentTime, output);
        } else if (e < i) {

            Internal(currentTime);
        } else {

            Internal(currentTime);
            External(currentTime, output);


        }
        Iterator<Job> value = completeQ.iterator();


    } while (completeQ.size() != numOfJob);
    DisplayFinalState(currentTime,output);//displaying the final state
    completeQ.clear();
}

input.close();
output.close();
}
public static void External(int currentTime, PrintWriter output){

  if(!allJobQ.isEmpty()){
    if(allJobQ.peek().getID()== -1){

        allJobQ.poll();

            Display(currentTime,output);

    }else {
            Job job=allJobQ.poll();
            if(job.getMemory()>availableMem || job.getDev()>availbleDev){

                holdQ2.add(job);
```

```
                    job.setStartTimeInReady(currentTime);//start timer when a job enter
the holdQ2

                }else if(job.getMemory()<=availableMem &&
job.getDev()<=availbleDev){

                    holdQ1.add(job);
                    availableMem-=job.getMemory();
                    availbleDev-=job.getDev();

                    update_SR_AR();
//
//
                }
            }
        }
    }
    public static void Internal(int CurrentTime) {

        getOutOfCPU(CurrentTime);


        if (!holdQ1.isEmpty()) {

            update_SR_AR();
            Quantum=ar;
            getIntoCPU(holdQ1.poll(), CurrentTime);
        }

    }
    public static void update_SR_AR() {
        if (holdQ1.isEmpty()) {//if no element , to avoid dividing over 0
            ar = sr = 0;
            return;
        }

        Iterator<Job> value = holdQ1.iterator();
        Job o ;


        while (value.hasNext()) {
             o = value.next();
            if (o.getP() == 1) {
                sr += (o.getR()-o.getAccuredTime()) * 2;
            } else {
```

```
            sr += (o.getR()-o.getAccuredTime()) * 1;
          }

        }


        ar = sr / (holdQ1.size());

        Quantum=ar;
        sr=0;
    }
    public static void Update_OUT_Hold(int currentTime){
        Iterator<Job> value = holdQ2.iterator();
        Job o ;


        double avgWT;
        double sumWT=0;
        while (value.hasNext()) {
            o = value.next();

              o.setFinishTimeOutOfReady(currentTime);//set finish time in holdQ2

              o.setPwt(o.getFinishTimeOutOfReady()-o.getStartTimeInReady());//set
proccess waiting time

              sumWT+=o.getPwt();

            }

             avgWT=sumWT/holdQ2.size();


         value = holdQ2.iterator();
        while (value.hasNext()) {
            o = value.next();
            if(o.getPpr()==-1){
                o.setPpr(o.getP());

            }else{
              if(o.getPwt()-avgWT>0){
             double DP = (o.getPwt()-avgWT)*0.2+o.getPpr()*0.8;
             o.setPpr(DP);

            }
```

```
        }

    }



}
public static void Get_OUT_Hold(int currentTime) {

    if(!holdQ2.isEmpty()){

        Update_OUT_Hold(currentTime);

   Job [] sortingArray= new Job[holdQ2.size()];
   Job o;
    Iterator<Job> value =holdQ2.iterator();
    while (value.hasNext()){

      o=value.next();

    }
     value =holdQ1.iterator();
    while (value.hasNext()){//copy element to array

      o=value.next();

    }
    CopyQueueToArray(sortingArray , holdQ2);

        for (int j = 0; j < sortingArray.length; j++) {

        }
    sortArray(1,sortingArray, holdQ2);
    holdQ2.clear();
        for (int j = 0; j < sortingArray.length; j++) {

            holdQ2.add(sortingArray[j]);
        }
        int sizeq2=holdQ2.size();
   for(int i=0;i<sizeq2;i++){
       Job temp=holdQ2.poll();
        if(temp.getMemory()<=availableMem && temp.getDev()<=availbleDev){
```

```
            holdQ1.add(temp);
            availableMem-=temp.getMemory();
            availbleDev-=temp.getDev();

           update_SR_AR();
        }else{
            holdQ2.add(temp);

        }


    }



    }
}
public static void CopyQueueToArray(Job arr[] , Queue<Job> h){

    int k =0;
    Job o;
    Iterator<Job> value = h.iterator();
    while (value.hasNext()){

      o=value.next();
       arr[k]=o;

      k++;
    }

}
public static void sortArray(int type,Job arr[], Queue<Job> h) {

    Job temp;
   if(type==1){
       for (int i = 0; i < arr.length; i++)
    {
        for (int j = i+ 1; j < arr.length; j++)
        {

            if (arr[i].getPpr() < arr[j].getPpr())
            {
                temp = arr[i];
                arr[i] = arr[j];
```

```
                arr[j] = temp;
            }
            else if(arr[i].getPpr() == arr[j].getPpr()){//if same jobs have the same ppr
then check the IDs
                if (arr[i].getID() > arr[j].getID()) {

                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;

            }
        }
    }
    }
    }else{
        for (int i = 0; i < arr.length; i++)
    {
        for (int j = i + 1; j < arr.length; j++)
        {
            if (arr[i].getID() > arr[j].getID())
            {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
    }

    h.clear();
    for (int j = 0; j < arr.length; j++) {

        h.add(arr[j]);
    }

}

public static void getIntoCPU(Job job,int currentTime){
cpu.setJobs(job);

cpu.setFree(false);

cpu.JInCPU.setStartTime(currentTime);
int RemainingTime=cpu.JInCPU.getR()-cpu.JInCPU.getAccuredTime();//set
remaining time
```

```
    if(RemainingTime>= Quantum){
        cpu.JInCPU.setFinishTime(currentTime+Quantum);
    }else{

        cpu.JInCPU.setFinishTime(currentTime+RemainingTime);
    }


    }

    public static void getOutOfCPU(int CurrentTime){

        if (cpu.isFree()== false) {



cpu.JInCPU.setAccuredTime(cpu.JInCPU.getAccuredTime()+(cpu.JInCPU.getFinis
hTime() - cpu.JInCPU.getStartTime()));

            if (cpu.JInCPU.getAccuredTime() == cpu.JInCPU.getR()) {
                availableMem += cpu.JInCPU.getMemory();
                availbleDev += cpu.JInCPU.getDev();


                completeQ.add(cpu.JInCPU);

                cpu.setFree(true);

                cpu.JInCPU=null;
                if(!holdQ2.isEmpty()){//when the job left the cpu it release its memory
and devices so we can move job from holdQ2 to holdQ1


                    Get_OUT_Hold(CurrentTime);
                }
            }else {//if the process not terminated


                Quantum=ar;
                holdQ1.add(cpu.JInCPU);

                cpu.setFree(true);
                update_SR_AR();
```

```
        }

    }
}
    public static void DisplayFinalState(int currentTime,PrintWriter output){
                        //print all the system
                    output.println("<< Final state of system: ");
                    output.println("  Current Available Main Memory =
"+availableMem);
                    output.println("  Current Devices             = "+availbleDev+"\n");
                    output.println("  Completed jobs:");
                    output.println("  ----------------");
                    output.println("  Job ID   Arrival Time    Finish Time  Turnaround
Time ");
                    output.println("
==================================================================
====");
                    Job [] sortingArray= new Job[completeQ.size()];
                    CopyQueueToArray(sortingArray , completeQ);
                    sortArray(2,sortingArray, completeQ);
                    completeQ.clear();
        for (int j = 0; j < sortingArray.length; j++) {

            completeQ.add(sortingArray[j]);
        }
        Iterator<Job> value = completeQ.iterator();
        double turaround=0;
        value = completeQ.iterator();
      while (value.hasNext()){
       Job job = value.next();
        output.printf("%5d %10d %15d %15d \n", job.getID(), job.getArrT(),
job.getFinishTime(), (job.getFinishTime()-job.getArrT()));
         turaround+=job.getFinishTime()-job.getArrT();
        }

                    output.println("");

                    output.println("");
                    output.printf("System Turnaround Time =
%.3f\n",turaround/completeQ.size());

output.println("\n*****************************************************************
****\n");
```

```
   }
 public static void Display(int CurrentTime, PrintWriter output) {
      output.println("<< At time " + CurrentTime + ": ");
      output.println("  Current Available Main Memory =" + availableMem );
      output.println("  Current Devices              =" + availbleDev);
      output.println("");
      output.println(" Completed jobs:");
      output.println(" ----------------");
      output.println("  Job ID   Arrival Time    Finish Time  Turnaround Time  ");
      output.println("
==================================================================
===");
       Job [] sortingArray1= new Job[completeQ.size()];
                    CopyQueueToArray(sortingArray1 , completeQ);
                    sortArray(2,sortingArray1, completeQ);
                    completeQ.clear();
         for (int j = 0; j < sortingArray1.length; j++) {

            completeQ.add(sortingArray1[j]);
         }
      Queue<Job> Temp = new LinkedList();
       Iterator<Job> value = completeQ.iterator();

     value = completeQ.iterator();
     while (value.hasNext()){
       Job job = value.next();
         output.printf("%5d %10d %13d %15d \n", job.getID(), job.getArrT(),
job.getFinishTime(), (job.getFinishTime()-job.getArrT()));

         }


      output.println("\n");
      output.println("  Hold Queue 2: ");
      output.println( "  -------------");
      Job [] sortingArray= new Job[holdQ2.size()];
                    CopyQueueToArray(sortingArray , holdQ2);
                    sortArray(2,sortingArray, holdQ2);
                    holdQ2.clear();
         for (int j = 0; j < sortingArray.length; j++) {

            holdQ2.add(sortingArray[j]);
         }
      value = holdQ2.iterator();
     while (value.hasNext()){
```

```
        Job job = value.next();
          output.printf(" %3d  ", job.getID());
          }
      output.println("");
      output.println("");
      output.println("");
      output.println("  Hold Queue 1 (Ready Queue):");
      output.println( "  ----------------");
      output.println("  JobID    NeedTime    Total Execution Time ");
      output.println(" ================================");value =
holdQ1.iterator();
       value = holdQ1.iterator();
     while (value.hasNext()){
        Job job = value.next();
          output.printf("%5d %10d %10d\n", job.getID(), (job.getR()-
job.getAccuredTime()),  job.getAccuredTime());//run time, time accrued
          }
      output.println("");
      output.println("");

      output.println("  Process running on the CPU: ");
      output.println("  ---------------------------");
      output.println("  Job ID   NeedTime    Total Execution Time");
      output.printf("%4d %8d %10d \n", cpu.JInCPU.getID(), (cpu.JInCPU.getR()-
cpu.JInCPU.getAccuredTime()),  cpu.JInCPU.getAccuredTime());//run time, time
accrued
      output.println("");
      output.println("");


   }

}
```

Job

```
public class Job {
private String type;
private int ID;
private int ArrT;
private int memory;
private int dev;
private int R;
private int p;
private double ppr;
private int finishTimeOutOfReady;
 private int startTimeInReady;
```

```
 private double pwt;
 private int finishTime;
 private int startTime;
 private int accuredTime;
 //private boolean PToSort;
private int turnArround=0;

public Job(int ArrT, int ID,int memory,int dev,int R,int p)   {
 this.ArrT=ArrT;
 this.ID=ID;
 this.memory= memory;
 this.dev=dev;
 this.p=p;
 this.R=R;
 this.startTime=0;
 this.finishTime=0;
 this.accuredTime=0;
 this.ppr=-1;
 //this.PToSort=true;
 this.turnArround=this.finishTime-this.ArrT;
 this.pwt=0;
}

   public int getTurnArround() {
       return turnArround;
   }

   public void setTurnArround(int turnArround) {
       this.turnArround = turnArround;
   }


//
//    public void setPToSort(boolean PToSort) {
//        this.PToSort = PToSort;
//    }
//    public boolean getPToSort() {
//        return PToSort;
//    }
   public void setPpr(double ppr) {
       this.ppr = ppr;
   }

   public double getPpr() {
       return ppr;
```

```
}

public int getFinishTimeOutOfReady() {
    return finishTimeOutOfReady;
}

public int getStartTimeInReady() {
    return startTimeInReady;
}

public double getPwt() {
    return pwt;
}

public void setFinishTimeOutOfReady(int finishTimeOutOfReady) {
    this.finishTimeOutOfReady = finishTimeOutOfReady;
}

public void setStartTimeInReady(int startTimeInReady) {
    this.startTimeInReady = startTimeInReady;
}

public void setPwt(double pwt) {
    this.pwt = pwt;
}

public int getFinishTime() {
    return finishTime;
}

public int getStartTime() {
    return startTime;
}

public void setFinishTime(int finishTime) {
    this.finishTime = finishTime;
}

public String getType() {
    return type;
}

public void setType(String type) {
    this.type = type;
}
```

```
public int getID() {
    return ID;
}

public void setID(int ID) {
    this.ID = ID;
}

public int getArrT() {
    return ArrT;
}

public void setArrT(int ArrT) {
    this.ArrT = ArrT;
}

public int getMemory() {
    return memory;
}

public void setMemory(int memory) {
    this.memory = memory;
}

public int getDev() {
    return dev;
}

public void setDev(int dev) {
    this.dev = dev;
}

public int getR() {
    return R;
}

public void setR(int R) {
    this.R = R;
}

public int getP() {
    return p;
}
```

```
    public void setP(int p) {
        this.p = p;
    }
    public void setStartTime(int startTime) {
        this.startTime = startTime;
    }
    public int getAccuredTime() {
        return accuredTime;
    }

    public void setAccuredTime(int accuredTime) {
        this.accuredTime = accuredTime;
    }
}
```

CPU

```
public class CPU {
  Job JInCPU;

  boolean free;

    public CPU(Job JInCPU, boolean free) {
        this.JInCPU = JInCPU;
        this.free = free;
    }



    public Job getJobs() {
        return JInCPU;
    }

    public void setJobs(Job jobs) {
        this.JInCPU = jobs;
    }

    public boolean isFree() {
        return free;
    }

    public void setFree(boolean free) {
        this.free = free;
    }
```

```
}
```

osStaticMain (Static)

```
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;
//Aya Kazzaz,Hadeel aloufi,Salwa Abbara,Mona Hafez,Aisha
Baskran,windows10,version1909(OS Build 18363.1256)
//processor:Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 2.00 GHz
//Installed memory(RAM): 8.00 GB(7.90 GB usable)
//compiler name and version:java version "1.8.0_191" ,Java(TM) SE Runtime
Environment


/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author WinDows
 */
public class osStaticMain {
 static int memorySize;
static int devices;
static int time;
static Queue<Job>allJobQ=new LinkedList();
static Queue<Job>holdQ1=new LinkedList();
static Queue<Job>holdQ2=new LinkedList();
static Queue<Job>sortedHoldQ2=new LinkedList();

static CPU cpu=new CPU(null,true);
static Queue<Job>completeQ=new LinkedList();
static int cStartTime=0;
static int availableMem=0;
static int availbleDev=0;
static int cThrehold=0;
static int numOfJob=0;
static int Quantum=5+4;
```

```
public static int i;
public static int e;


    public static void main(String[] args) throws FileNotFoundException {
    java.io.File file = new java.io.File("input1.txt");
   // java.io.File file = new java.io.File("input2.txt");you can change between these 3
input files to check outputs files
    //java.io.File file = new java.io.File("input3.txt");
    java.io.File outputFile = new java.io.File("output.txt");
    PrintWriter output=new PrintWriter(outputFile);//print writer to write to the
outputfile
    Scanner input=new Scanner(file);
    while (input.hasNext()) {

     String command1=input.next();

     time=input.nextInt();

     memorySize=Integer.parseInt(input.next().substring(2));
     availableMem=memorySize;

     devices=Integer.parseInt(input.next().substring(2));

       availbleDev=devices;
     Job job;
     numOfJob=0;
     int number = 0;
     do{

         String command2=input.next();

         if(command2.equals("A")){
           int ArrT= Integer.parseInt(input.next());

           int ID=Integer.parseInt(input.next().substring(2));

           int memory=Integer.parseInt(input.next().substring(2));

           int dev=Integer.parseInt(input.next().substring(2));

           int R=Integer.parseInt(input.next().substring(2));

           int p=Integer.parseInt(input.next().substring(2));
```

```
  if(memory<=memorySize&&dev<=devices){
   job=new Job(ArrT,ID,memory,dev,R,p);

   allJobQ.add(job);
   numOfJob++;//counter to count the number of jobs

  }
}else if(command2.equals("D")){

   int time1=input.nextInt();

   if(time1<999999){
    job=new Job(time1,-1,-1,-1,-1,-1);

    allJobQ.add(job);
   }
   else{

   break;
}
}
} while(input.hasNext());
  Job j=allJobQ.poll();

  int currentTime=j.getArrT();

  availableMem-=j.getMemory();

  getIntoCPU(j,currentTime);

  i = 0;
   e = 0;

   do {

      if (!(allJobQ.isEmpty())) {

         i = allJobQ.peek().getArrT();
      } else {
         i = Integer.MAX_VALUE;
      }

      if (cpu.isFree()==false) {
```

```
            e = cpu.JInCPU.getFinishTime();

        } else {
            e = Integer.MAX_VALUE;
        }
        currentTime = Math.min(i, e);

        if (i < e) {

            External(currentTime, output);
        } else if (e < i) {

            Internal(currentTime);
        } else {

            Internal(currentTime);
            External(currentTime, output);


        }

    } while (completeQ.size() != numOfJob);
    DisplayFinalState(currentTime,output);//displaying the final state
    completeQ.clear();
}

input.close();
output.close();
}
public static void External(int currentTime, PrintWriter output){

  if(!allJobQ.isEmpty()){
    if(allJobQ.peek().getID()== -1){

        allJobQ.poll();

            Display(currentTime,output);

    }else {
            Job job=allJobQ.poll();
            if(job.getMemory()>availableMem || job.getDev()>availbleDev){

                holdQ2.add(job);
```

```
                }else if(job.getMemory()<=availableMem &&
job.getDev()<=availbleDev){

                    holdQ1.add(job);
                    availableMem-=job.getMemory();
                    availbleDev-=job.getDev();


                }
            }
        }
    }
    public static void Internal(int CurrentTime) {

        getOutOfCPU(CurrentTime);

        if (!holdQ1.isEmpty()) {

            getIntoCPU(holdQ1.poll(), CurrentTime);
        }

    }


    public static void Get_OUT_Hold(int currentTime) {
//timer start
        if(!holdQ2.isEmpty()){

       Job [] sortingArray= new Job[holdQ2.size()];
       Job o;
        Iterator<Job> value =holdQ2.iterator();
        while (value.hasNext()){

          o=value.next();

        }
         value =holdQ1.iterator();
        while (value.hasNext()){

          o=value.next();

        }
        CopyQueueToArray(sortingArray , holdQ2);
```

```
        for (int j = 0; j < sortingArray.length; j++) {

        }

    sortArray(2,sortingArray, holdQ2);
    holdQ2.clear();
        for (int j = 0; j < sortingArray.length; j++) {


            holdQ2.add(sortingArray[j]);
        }
   int sizeq2=holdQ2.size();
   for(int i=0;i<sizeq2;i++){
      Job temp=holdQ2.poll();
       if(temp.getMemory()<=availableMem && temp.getDev()<=availbleDev){

            holdQ1.add(temp);
            availableMem-=temp.getMemory();
                  availbleDev-=temp.getDev();

        }else{
            holdQ2.add(temp);

        }



    }




    }
}
public static void CopyQueueToArray(Job arr[] , Queue<Job> h){

    int k =0;
    Job o;
    Iterator<Job> value = h.iterator();
    while (value.hasNext()){

      o=value.next();
       arr[k]=o;

      k++;
    }
```

```
    }
    public static void sortArray(int type, Job arr[], Queue<Job> h) {

      Job temp;
     if(type==1){
        for (int i = 0; i < arr.length; i++)
      {
         for (int j = i + 1; j < arr.length; j++)
         {
            if (arr[i].getID() > arr[j].getID())
            {
               temp = arr[i];
               arr[i] = arr[j];
               arr[j] = temp;
            }
         }
      }
     }else{
        for (int i = 0; i < arr.length; i++)
      {
         for (int j = i + 1; j < arr.length; j++)
         {
            if (arr[i].getP() < arr[j].getP())
            {
               temp = arr[i];
               arr[i] = arr[j];
               arr[j] = temp;
            }else if(arr[i].getP() == arr[j].getP()){//if same jobs have the same priority
then check the IDs
               if (arr[i].getID() > arr[j].getID())
            {
               temp = arr[i];
               arr[i] = arr[j];
               arr[j] = temp;
            }
            }
         }
      }
     }

      h.clear();
      for (int j = 0; j < arr.length; j++) {
```

```
            h.add(arr[j]);
        }

    }

    public static void getIntoCPU(Job job,int currentTime){
    cpu.setJobs(job);

    cpu.setFree(false);

    cpu.JInCPU.setStartTime(currentTime);

    int RemainingTime=cpu.JInCPU.getR()-cpu.JInCPU.getAccuredTime();

    if(RemainingTime>= Quantum){
        cpu.JInCPU.setFinishTime(currentTime+Quantum);
    }else{

        cpu.JInCPU.setFinishTime(currentTime+RemainingTime);
    }


    }

    public static void getOutOfCPU(int CurrentTime){

        if (cpu.isFree()== false) {


cpu.JInCPU.setAccuredTime(cpu.JInCPU.getAccuredTime()+(cpu.JInCPU.getFinishT
ime() - cpu.JInCPU.getStartTime()));


            if (cpu.JInCPU.getAccuredTime() == cpu.JInCPU.getR()) {
                availableMem += cpu.JInCPU.getMemory();
                availbleDev += cpu.JInCPU.getDev();


                completeQ.add(cpu.JInCPU);

                cpu.setFree(true);

                cpu.JInCPU=null;
```

```
            if(!holdQ2.isEmpty()){//when the job left the cpu it release its memory and
devices so we can move job from holdQ2 to holdQ1

                Get_OUT_Hold(CurrentTime);
            }
        }else {//if the process not terminated


            holdQ1.add(cpu.JInCPU);

            cpu.setFree(true);



        }

    }
}
    public static void DisplayFinalState(int currentTime,PrintWriter output){
                        //print all the system
                    output.println("<< Final state of system: ");
                    output.println("  Current Available Main Memory =
"+availableMem);
                    output.println("  Current Devices             = "+availbleDev);
                    output.println("");
                    output.println("  Completed jobs:");
                    output.println("  ----------------");
                    output.println("  Job ID   Arrival Time    Finish Time  Turnaround
Time ");
                    output.println("
==================================================================
=");
                    Job [] sortingArray= new Job[completeQ.size()];
                    CopyQueueToArray(sortingArray , completeQ);
                    sortArray(1,sortingArray, completeQ);
                    completeQ.clear();
        for (int j = 0; j < sortingArray.length; j++) {
            //sortingArray[j].setPToSort(true);
            //System.out.println("Array---------------"+sortingArray[j].getID());
            completeQ.add(sortingArray[j]);
        }
        Iterator<Job> value = completeQ.iterator();
         double turaround=0;
        value = completeQ.iterator();
```

```
      while (value.hasNext()){
        Job job = value.next();
          output.printf("%5d %10d %15d %15d \n", job.getID(), job.getArrT(),
job.getFinishTime(), (job.getFinishTime()-job.getArrT()));
          turaround+=job.getFinishTime()-job.getArrT();

          }
                    //printJobscompleteQue(sortingArray);
                   // printJobscompleteQue();
                    output.println("");
                    output.println("");
                    output.printf("System Turnaround Time =
%.3f\n",turaround/completeQ.size());

output.println("\n*****************************************************************\n");

   }
 public static void Display(int CurrentTime, PrintWriter output) {


    output.println("<< At time " + CurrentTime + ": ");
      output.println("  Current Available Main Memory =" + availableMem );
      output.println("  Current Devices              = " + availbleDev);
      output.println("");
      output.println(" Completed jobs:");
      output.println("  ----------------");
      output.println("  Job ID   Arrival Time    Finish Time  Turnaround Time  ");
      output.println("
==============================================================
==");
      // Queue<Job> Temp = new LinkedList();
      Job [] sortingArray1= new Job[completeQ.size()];
                    CopyQueueToArray(sortingArray1 , completeQ);
                    sortArray(1,sortingArray1, completeQ);
                    completeQ.clear();
          for (int j = 0; j < sortingArray1.length; j++) {

             completeQ.add(sortingArray1[j]);
          }

        Iterator<Job> value = completeQ.iterator();

      value = completeQ.iterator();
      while (value.hasNext()){
```

```
        Job job = value.next();
          output.printf("%5d %10d %13d %15d \n", job.getID(), job.getArrT(),
job.getFinishTime(), (job.getFinishTime()-job.getArrT()));

          }

      ////////////////////////////////////
      output.println("\n");
      output.println("  Hold Queue 2: ");
      output.println( "  --------------");
      Job [] sortingArray= new Job[holdQ2.size()];
                    CopyQueueToArray(sortingArray , holdQ2);
                    sortArray(1,sortingArray, holdQ2);
                    holdQ2.clear();
          for (int j = 0; j < sortingArray.length; j++) {

              holdQ2.add(sortingArray[j]);
          }
      value = holdQ2.iterator();
     while (value.hasNext()){
        Job job = value.next();
          output.printf(" %3d  ", job.getID());
          }

      output.println("");
      output.println("");
      output.println("");
      output.println("  Hold Queue 1 (Ready Queue): ");
      output.println( "  ----------------");
      output.println("  JobID    NeedTime    Total Execution Time ");
      output.println(" ================================");value =
holdQ1.iterator();
       value = holdQ1.iterator();
     while (value.hasNext()){
        Job job = value.next();
          output.printf("%5d %10d %10d\n", job.getID(), (job.getR()-
job.getAccuredTime()), job.getAccuredTime());//run time, time accrued
          }
      output.println("");
      output.println("");
      output.println("  Process running on the CPU: ");
      output.println("  ----------------------------");
      output.println("  Job ID   NeedTime    Total Execution Time");
```

```
        output.printf("%5d %10d %10d \n", cpu.JInCPU.getID(), (cpu.JInCPU.getR()-
cpu.JInCPU.getAccuredTime()), cpu.JInCPU.getAccuredTime());//run time, time
accrued
        output.println("");
        output.println("");
    }


}
```

Job

```
public class Job {
private String type;
private int ID;
private int ArrT;
private int memory;
private int dev;
private int R;
private int p;
private double ppr;
private int finishTimeOutOfReady;
 private int startTimeInReady;
 private double pwt;
 private int finishTime;
 private int startTime;
 private int accuredTime;
 //private boolean PToSort;
private int turnArround=0;

public Job(int ArrT, int ID,int memory,int dev,int R,int p)   {
 this.ArrT=ArrT;
 this.ID=ID;
 this.memory= memory;
 this.dev=dev;
 this.p=p;
 this.R=R;
 this.startTime=0;
 this.finishTime=0;
 this.accuredTime=0;
 this.ppr=-1;
 //this.PToSort=true;
 this.turnArround=this.finishTime-this.ArrT;
 this.pwt=0;
}
```

```
    public int getTurnArround() {
        return turnArround;
    }

    public void setTurnArround(int turnArround) {
        this.turnArround = turnArround;
    }


//
//    public void setPToSort(boolean PToSort) {
//        this.PToSort = PToSort;
//    }
//    public boolean getPToSort() {
//        return PToSort;
//    }
    public void setPpr(double ppr) {
        this.ppr = ppr;
    }

    public double getPpr() {
        return ppr;
    }

    public int getFinishTimeOutOfReady() {
        return finishTimeOutOfReady;
    }

    public int getStartTimeInReady() {
        return startTimeInReady;
    }

    public double getPwt() {
        return pwt;
    }

    public void setFinishTimeOutOfReady(int finishTimeOutOfReady) {
        this.finishTimeOutOfReady = finishTimeOutOfReady;
    }

    public void setStartTimeInReady(int startTimeInReady) {
        this.startTimeInReady = startTimeInReady;
    }
```

```
public void setPwt(double pwt) {
    this.pwt = pwt;
}

public int getFinishTime() {
    return finishTime;
}

public int getStartTime() {
    return startTime;
}

public void setFinishTime(int finishTime) {
    this.finishTime = finishTime;
}

public String getType() {
    return type;
}

public void setType(String type) {
    this.type = type;
}

public int getID() {
    return ID;
}

public void setID(int ID) {
    this.ID = ID;
}

public int getArrT() {
    return ArrT;
}

public void setArrT(int ArrT) {
    this.ArrT = ArrT;
}

public int getMemory() {
    return memory;
}
```

```
    public void setMemory(int memory) {
        this.memory = memory;
    }

    public int getDev() {
        return dev;
    }

    public void setDev(int dev) {
        this.dev = dev;
    }

    public int getR() {
        return R;
    }

    public void setR(int R) {
        this.R = R;
    }

    public int getP() {
        return p;
    }

    public void setP(int p) {
        this.p = p;
    }
    public void setStartTime(int startTime) {
        this.startTime = startTime;
    }
    public int getAccuredTime() {
        return accuredTime;
    }

    public void setAccuredTime(int accuredTime) {
        this.accuredTime = accuredTime;
    }
}
```

CPU

```
public class CPU {
  Job JInCPU;
// private int Quantum;
```

```
  boolean free;

  public CPU(Job JInCPU, boolean free) {
      this.JInCPU = JInCPU;
      this.free = free;
  }

  public Job getJobs() {
      return JInCPU;
  }

  public void setJobs(Job jobs) {
      this.JInCPU = jobs;
  }

  public boolean isFree() {
      return free;
  }

  public void setFree(boolean free) {
      this.free = free;
  }


}
```

4. A copy of the program output on each test data file

Screenshots for dynamic roundrobin

- ### Input 1

```
<< Final state of system:
  Current Available Main Memory = 120
  Current Devices               = 0

  Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1         1              11             10
    2         2              51             49
    3         3              31             28
    4         5              62             57
    5         7              92             85
    6         9              74             65
    7        10              50             40
    8        12             117            105
   10        20             102             82
   11        26              60             34
   12        45              77             32
   13        58              82             24


System Turnaround Time =  50.917

*******************************************************************

<< Final state of system:
  Current Available Main Memory = 40
  Current Devices               = 0

  Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1         2              17             15
    2         3              24             21
```

```
System Turnaround Time =  50.917

*******************************************************************

<< Final state of system:
  Current Available Main Memory = 40
  Current Devices               = 0

  Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1         2              17             15
    2         3              24             21
    3         4              56             52
    4         6              29             23
    5         9              64             55
    6        12              84             72
    7        15              46             31
    8        17              95             78
   10        22             157            135
   11        24             176            152
   12        27             145            118
   13        29             143            114
   14        33             191            158
   15        35             142            107


System Turnaround Time =  80.786

*******************************************************************
```

- ## Input 2

```
<< At time 16:
  Current Available Main Memory =56
  Current Devices               =1

 Completed jobs:
 ---------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1         3             6              3
    2         4             8              4
    3         6             9              3
    4         8            14              6


  Hold Queue 2:
  -------------
   7     8


  Hold Queue 1 (Ready Queue):
  ---------------
  JobID    NeedTime    Total Execution Time
 ================================


  Process running on the CPU:
  ---------------------------
  Job ID   NeedTime    Total Execution Time
   6        7          0


<< Final state of system:
  Current Available Main Memory = 100
  Current Devices               = 1

  Completed jobs:
```

```
<< Final state of system:
  Current Available Main Memory = 100
  Current Devices               = 1

  Completed jobs:
  ---------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1         3              6              3
    2         4              8              4
    3         6              9              3
    4         8             14              6
    6        12             21              9
    7        13             31             18
    8        15             35             20
    9        18             30             12
   10        20             40             20
   12        24             50             26
   13        25             55             30
   14        28             44             16
   15        31             47             16


System Turnaround Time =  14.077

*******************************************************************

<< At time 9:
  Current Available Main Memory =160
  Current Devices               =1

 Completed jobs:
 ---------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
```

output - Notepad

File Edit Format View Help

```
*****************************************************************

<< At time 9:
  Current Available Main Memory =160
  Current Devices               =1

 Completed jobs:
 ---------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1          2             7               5


  Hold Queue 2:
  -------------


  Hold Queue 1 (Ready Queue):
  ---------------
  JobID    NeedTime    Total Execution Time
 ================================
    3          2          0
    4          9          0
    5          5          0


  Process running on the CPU:
  ---------------------------
  Job ID   NeedTime    Total Execution Time
   2        4          0


<< Final state of system:
  Current Available Main Memory = 200
  Current Devices               = 1
```

Activate Windows
Go to Settings to activate Windows.

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 11:54 PM 12/12/2020

output - Notepad

File Edit Format View Help

```
  ---------------------------
  Job ID   NeedTime    Total Execution Time
   2        4          0


<< Final state of system:
  Current Available Main Memory = 200
  Current Devices               = 1

  Completed jobs:
  ---------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1          2              7               5
    2          4             11               7
    3          5             13               8
    4          7             22              15
    5          8             27              19
    6         10             31              21
    7         12             35              23
    8         14             36              22
    9         15             40              25
   10         16             47              31
   11         18             50              32
   12         20             53              33
   13         22             59              37
   14         24             62              38
   15         27             65              38


System Turnaround Time =  23.600

*****************************************************************
```

Activate Windows
Go to Settings to activate Windows.

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 11:55 PM 12/12/2020

- # Input 3

```
<< At time 14:
  Current Available Main Memory =56
  Current Devices               =1

 Completed jobs:
 ---------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1         3             6              3
    2         4             8              4
    3         6             9              3
    4         8            14              6


  Hold Queue 2:
  -------------
   7


  Hold Queue 1 (Ready Queue):
  ---------------
  JobID    NeedTime    Total Execution Time
 ================================


  Process running on the CPU:
  ---------------------------
  Job ID   NeedTime    Total Execution Time
   6        3          0


<< Final state of system:
  Current Available Main Memory = 100
  Current Devices               = 1

  Completed jobs:
```

```
<< Final state of system:
  Current Available Main Memory = 100
  Current Devices               = 1

  Completed jobs:
  ---------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1         3              6              3
    2         4              8              4
    3         6              9              3
    4         8             14              6
    6        12             17              5
    7        13             22              9
    8        15             21              6
    9        17             36             19
   10        20             27              7
   11        22             44             22
   12        23             47             24
   14        28             39             11
   15        29             42             13


System Turnaround Time =  10.154

*********************************************************************

<< At time 11:
  Current Available Main Memory =10
  Current Devices               =1

 Completed jobs:
 ---------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
```

output - Notepad

File Edit Format View Help

```
<< At time 11:
  Current Available Main Memory =10
  Current Devices               =1

 Completed jobs:
 ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1          2             7               5


  Hold Queue 2:
  --------------



  Hold Queue 1 (Ready Queue):
  ----------------
  JobID    NeedTime    Total Execution Time
 ==================================
    4          4          0
    5          5          0
    2          1          3
    6          4          0


  Process running on the CPU:
  ----------------------------
  Job ID   NeedTime    Total Execution Time
   3        2          0


<< Final state of system:
  Current Available Main Memory = 60
  Current Devices               = 1
```

Activate Windows
Go to Settings to activate Windows.

Ln 7, Col 58 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 11:57 PM 12/12/2020

output - Notepad

File Edit Format View Help

```
<< Final state of system:
  Current Available Main Memory = 60
  Current Devices               = 1

  Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1          2             7               5
    2          4            18              14
    3          5            12               7
    4          7            24              17
    5          9            27              18
    6         10            32              22
    7         12            34              22
    8         14            23               9
    9         15            41              26
   10         17            47              30
   11         18            48              30
   12         20            53              33
   13         22            57              35
   14         23            49              26
   15         25            52              27


System Turnaround Time =  21.400

*******************************************************************

<< At time 10:
  Current Available Main Memory =40
  Current Devices               =6

 Completed jobs:
```

Activate Windows
Go to Settings to activate Windows.

Ln 7, Col 58 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 11:58 PM 12/12/2020

output - Notepad

File Edit Format View Help

```
<< At time 10:
  Current Available Main Memory =40
  Current Devices               =6

 Completed jobs:
 ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1          2            10               8


  Hold Queue 2:
  --------------



  Hold Queue 1 (Ready Queue):
  ----------------
  JobID    NeedTime    Total Execution Time
 ==================================
    4          7          0
    3          8          0


  Process running on the CPU:
  ----------------------------
  Job ID   NeedTime    Total Execution Time
   2        8          0


<< At time 39:
  Current Available Main Memory =15
  Current Devices               =6

 Completed jobs:
 ----------------
```

Activate Windows
Go to Settings to activate Windows.

Ln 7, Col 58 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 11:58 PM 12/12/2020

output - Notepad

File Edit Format View Help

```
<< At time 39:
  Current Available Main Memory =15
  Current Devices               =6

 Completed jobs:
 ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1          2            10               8
    2          3            18              15
    3          4            33              29
    4          6            25              19


  Hold Queue 2:
  --------------
  7     8    10    11    16    17


  Hold Queue 1 (Ready Queue):
  ----------------
  JobID    NeedTime    Total Execution Time
 ================================
    9          3          0
   13          6          0
   15          1          0
    5          4          6


  Process running on the CPU:
  ----------------------------
  Job ID   NeedTime    Total Execution Time
   6        4          0
```

Activate Windows
Go to Settings to activate Windows.

Ln 7, Col 58 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 11:58 PM 12/12/2020

output - Notepad

File Edit Format View Help

```
  Process running on the CPU:
  ----------------------------
  Job ID   NeedTime    Total Execution Time
   6        4          0


<< Final state of system:
  Current Available Main Memory = 100
  Current Devices               = 6

  Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1          2              10               8
    2          3              18              15
    3          4              33              29
    4          6              25              19
    5         14              66              52
    6         15              43              28
    7         18              78              60
    8         20              83              63
    9         23              46              23
   10         25              58              33
   11         28              95              67
   13         30              60              30
   15         33              51              18
   16         35              76              41
   17         36              97              61
   19         41             107              66
   21         47             111              64
   22         51              65              14
   28         59             114              55


System Turnaround Time =  39.263
```

Activate Windows
Go to Settings to activate Windows.

Ln 7, Col 58 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 11:58 PM 12/12/2020

Screenshots for brute force round robin

- ## Input 1

```
<< Final state of system:
  Current Available Main Memory = 120
  Current Devices               = 0

  Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1          1              33             32
    2          2              40             38
    3          3              23             20
    4          5              56             51
    5          7             102             95
    6          9             100             91
    7         10              61             51
    8         12             117            105
   10         20              88             68
   11         26              70             44
   12         45              73             28
   13         58              78             20


System Turnaround Time =  53.583

*****************************************************************

<< Final state of system:
  Current Available Main Memory = 40
  Current Devices               = 0

  Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1          2              29             27
    2          3              17             14
```

```
System Turnaround Time =  53.583

*****************************************************************

<< Final state of system:
  Current Available Main Memory = 40
  Current Devices               = 0

  Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1          2              29             27
    2          3              17             14
    3          4             147            143
    4          6              22             16
    5          9              45             36
    6         12             170            158
    7         15              74             59
    8         17             158            141
   10         22              73             51
   11         24             191            167
   12         27             121             94
   13         29             188            159
   14         33             113             80
   15         35             120             85


System Turnaround Time =  87.857

*****************************************************************
```

- ## Input 2

```
<< At time 16:
  Current Available Main Memory =56
  Current Devices               = 1

 Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1         3             6              3
    2         4             8              4
    3         6             9              3
    4         8            14              6


  Hold Queue 2:
  --------------
   7     8


  Hold Queue 1 (Ready Queue):
  ----------------
  JobID    NeedTime    Total Execution Time
 ================================


  Process running on the CPU:
  ----------------------------
  Job ID   NeedTime    Total Execution Time
    6         7          0


<< Final state of system:
  Current Available Main Memory = 100
  Current Devices               = 1

  Completed jobs:
```

```
<< Final state of system:
  Current Available Main Memory = 100
  Current Devices               = 1

  Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1         3             6              3
    2         4             8              4
    3         6             9              3
    4         8            14              6
    6        12            21              9
    7        13            38             25
    8        15            42             27
    9        18            30             12
   10        20            47             27
   12        24            50             26
   13        25            55             30
   14        28            34              6
   15        31            37              6


System Turnaround Time =  14.154

*******************************************************************

<< At time 9:
  Current Available Main Memory =160
  Current Devices               = 1

 Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1         2             7              5
```

output - Notepad

File Edit Format View Help

```
*******************************************************************

<< At time 9:
  Current Available Main Memory =160
  Current Devices               = 1

 Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1          2             7               5


  Hold Queue 2:
  --------------



  Hold Queue 1 (Ready Queue):
  ----------------
  JobID    NeedTime    Total Execution Time
 ================================
    3          2          0
    4          9          0
    5          5          0


  Process running on the CPU:
  ----------------------------
  Job ID   NeedTime    Total Execution Time
    2          4          0


<< Final state of system:
  Current Available Main Memory = 200
  Current Devices               = 1
```

Activate Windows
Go to Settings to activate Windows.

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 12:05 AM 12/13/2020

output - Notepad

File Edit Format View Help

```
  ----------------------------
  Job ID   NeedTime    Total Execution Time
    2          4          0


<< Final state of system:
  Current Available Main Memory = 200
  Current Devices               = 1

  Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1          2              7               5
    2          4             11               7
    3          5             13               8
    4          7             53              46
    5          8             26              18
    6         10             30              20
    7         12             34              22
    8         14             35              21
    9         15             39              24
   10         16             46              30
   11         18             49              31
   12         20             52              32
   13         22             59              37
   14         24             62              38
   15         27             65              38


System Turnaround Time =  25.133

*******************************************************************
```

Activate Windows
Go to Settings to activate Windows.

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 12:05 AM 12/13/2020

- ## Input 3

```
<< At time 14:
  Current Available Main Memory =56
  Current Devices               = 1

 Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1         3             6              3
    2         4             8              4
    3         6             9              3
    4         8            14              6


  Hold Queue 2:
  --------------
   7


  Hold Queue 1 (Ready Queue):
  ----------------
  JobID    NeedTime    Total Execution Time
 ================================


  Process running on the CPU:
  ----------------------------
  Job ID   NeedTime    Total Execution Time
    6         3         0


<< Final state of system:
  Current Available Main Memory = 100
  Current Devices               = 1

  Completed jobs:
```

```
<< Final state of system:
  Current Available Main Memory = 100
  Current Devices               = 1

  Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
    1         3              6              3
    2         4              8              4
    3         6              9              3
    4         8             14              6
    6        12             17              5
    7        13             22              9
    8        15             21              6
    9        17             31             14
   10        20             36             16
   11        22             44             22
   12        23             47             24
   14        28             39             11
   15        29             42             13


System Turnaround Time =  10.462

*********************************************************************

<< At time 11:
  Current Available Main Memory =20
  Current Devices               = 1

 Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ================================================================
```

output - Notepad

File Edit Format View Help

```
*******************************************************************

<< At time 11:
  Current Available Main Memory =20
  Current Devices               = 1

 Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1          2             7              5
    2          4            11              7


  Hold Queue 2:
  --------------



  Hold Queue 1 (Ready Queue):
  ----------------
  JobID    NeedTime    Total Execution Time
 ================================
    4          4          0
    5          5          0
    6          4          0


  Process running on the CPU:
  ----------------------------
  Job ID   NeedTime    Total Execution Time
    3          2          0


<< Final state of system:
  Current Available Main Memory = 60
```

Activate Windows
Go to Settings to activate Windows.

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 12:07 AM 12/13/2020

output - Notepad

File Edit Format View Help

```
<< Final state of system:
  Current Available Main Memory = 60
  Current Devices               = 1

  Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1          2             7              5
    2          4            11              7
    3          5            13              8
    4          7            17             10
    5          9            22             13
    6         10            26             16
    7         12            30             18
    8         14            31             17
    9         15            35             20
   10         17            39             22
   11         18            42             24
   12         20            45             25
   13         22            51             29
   14         23            54             31
   15         25            57             32


System Turnaround Time =  18.467

*******************************************************************

<< At time 10:
  Current Available Main Memory =40
  Current Devices               = 6

 Completed jobs:
```

Activate Windows
Go to Settings to activate Windows.

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 12:08 AM 12/13/2020

output - Notepad

File Edit Format View Help

```
<< At time 10:
  Current Available Main Memory =40
  Current Devices               = 6

 Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1          2            10               8


  Hold Queue 2:
  --------------



  Hold Queue 1 (Ready Queue):
  ----------------
  JobID    NeedTime    Total Execution Time
 ================================
    4          7          0
    3          8          0


  Process running on the CPU:
  ----------------------------
  Job ID   NeedTime    Total Execution Time
    2          8          0


<< At time 39:
  Current Available Main Memory =15
  Current Devices               = 6

 Completed jobs:
```

Activate Windows
Go to Settings to activate Windows.

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 12:08 AM 12/13/2020

output - Notepad

File Edit Format View Help

```
<< At time 39:
  Current Available Main Memory =15
  Current Devices               = 6

 Completed jobs:
  ----------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1          2            10               8
    2          3            18              15
    3          4            33              29
    4          6            25              19


  Hold Queue 2:
  --------------
   7     8    10    11    16    17


  Hold Queue 1 (Ready Queue):
  ----------------
  JobID    NeedTime    Total Execution Time
 ================================
    6          4          0
    9          3          0
   13          6          0
   15          1          0


  Process running on the CPU:
  ----------------------------
  Job ID   NeedTime    Total Execution Time
    5         10          0
```

Activate Windows
Go to Settings to activate Windows.

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Type here to search | ENG | 12:08 AM 12/13/2020

output - Notepad

File Edit Format View Help

```
<< Final state of system:
  Current Available Main Memory = 100
  Current Devices               = 6

  Completed jobs:
  ---------------
  Job ID   Arrival Time    Finish Time  Turnaround Time
  ==================================================================
    1          2              10               8
    2          3              18              15
    3          4              33              29
    4          6              25              19
    5         14              57              43
    6         15              45              30
    7         18              78              60
    8         20              83              63
    9         23              48              25
   10         25              61              36
   11         28             114              86
   13         30              54              24
   15         33              55              22
   16         35              69              34
   17         36              85              49
   19         41              95              54
   21         47              99              52
   22         51              66              15
   28         59             102              43


System Turnaround Time =  37.211

*******************************************************************
```

Activate Windows
Go to Settings to activate Windows.

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

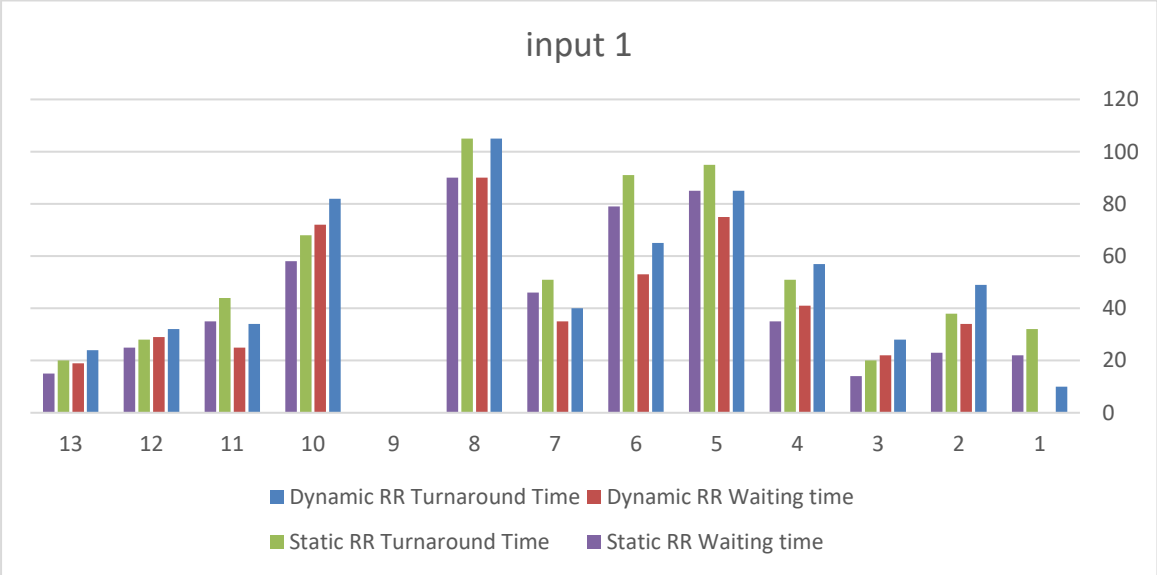
Type here to search | ENG | 12:08 AM 12/13/2020

5. The Comparative study of PART II

Input 1

Current Available Main Memory = 120, Current Device= 0

Dynamic RR		Static RR		
Turnaround Time	Waiting time	Turnaround Time	Waiting time	Job ID
10	0	32	22	1
49	34	38	23	2
28	22	20	14	3
57	41	51	35	4
85	75	95	85	5
65	53	91	79	6
40	35	51	46	7
105	90	105	90	8
it has no available memory.				9
82	72	68	58	10
34	25	44	35	11
32	29	28	25	12
24	19	20	15	13

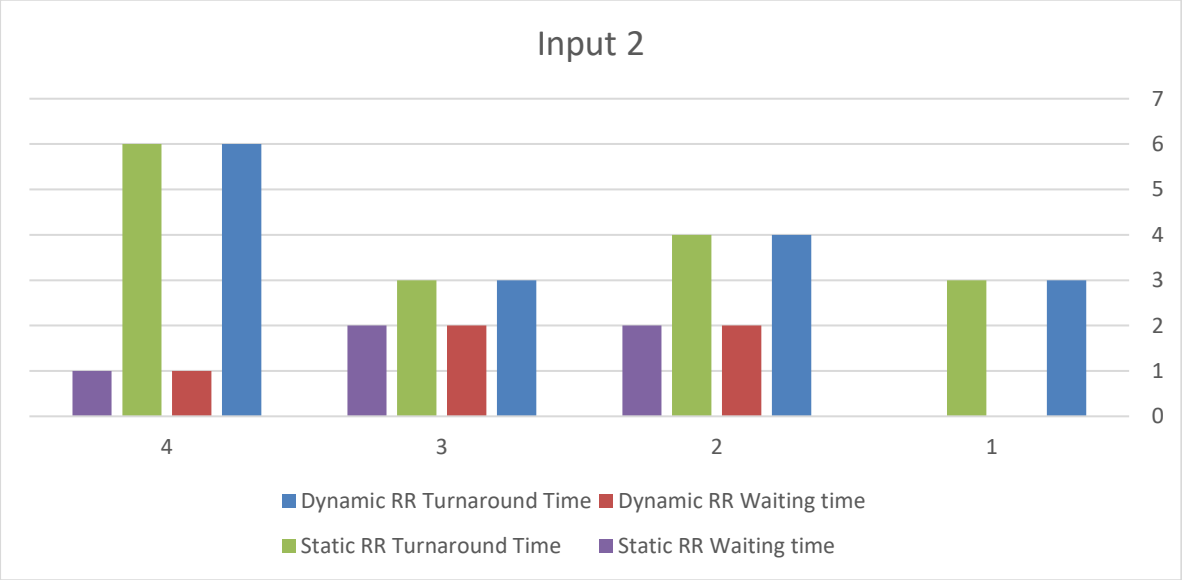


✓ **Here the memory stops at J=9 because its memory exceeds 120, it equal 130, so here the LOOP stops it.**

Input 2

Current Available Main Memory= 56, Current Device= 1

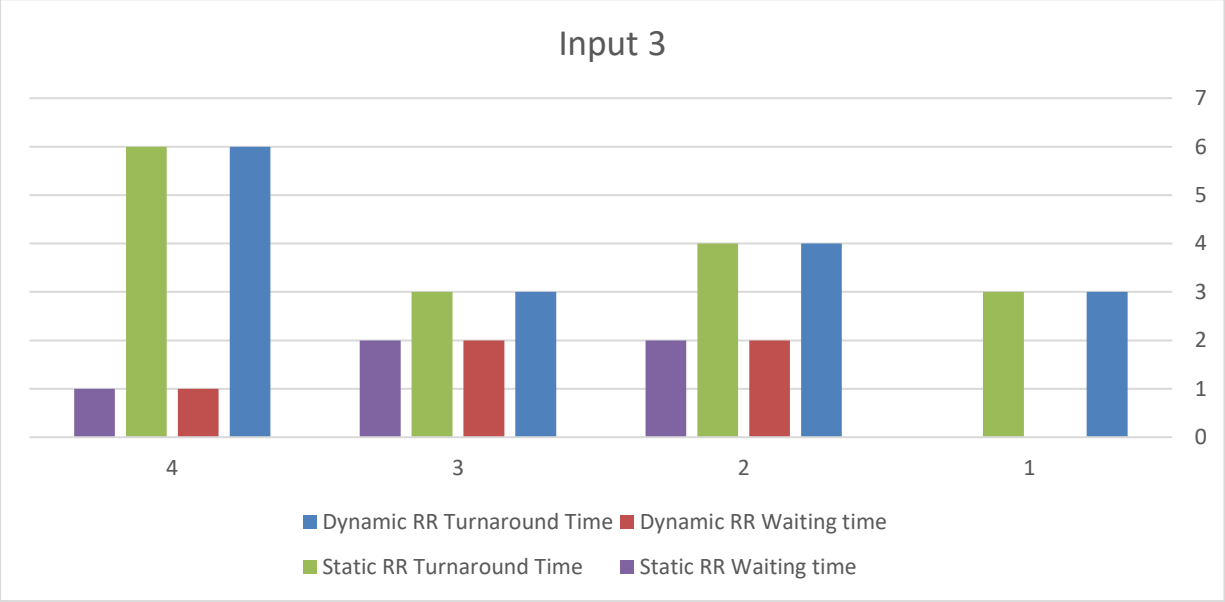
Dynamic RR		Static RR		
Turnaround Time	Waiting time	Turnaround Time	Waiting time	Job ID
3	0	3	0	1
4	2	4	2	2
3	2	3	2	3
6	1	6	1	4
it has no available memory.				5
The memory required in the J=5 equals 241, so this is not available for the existing memory.				6
				7
Everything that follows after the J=5 does not enter memory and is canceled.				8



Input 3

Current Available Main Memory= 56, Current Device= 1

Dynamic RR		**Static RR**		
Turnaround Time	Waiting time	Turnaround Time	Waiting time	Job ID
3	0	3	0	1
4	2	4	2	2
3	2	3	2	3
6	1	6	1	4
it has no available memory.				5
The memory required in the J=5 equals 241, so this is not available for the existing memory.				6
Everything that follows after the J=5 does not enter memory and is canceled.				7



- ✓ **When choosing a main memory of 56, both input 2&3 have the same turnaround time and waiting time in the Dynamic and Static RR.**
- ✓ **And for each of them, memory is not allowed at the same memory capacity in J=5, so both are equal to 241, so the memory stops then and the LOOp stops.**

	Waiting time	Turnaround time
Static RR	There is no Static time waiting only there is a priority for time.	It takes more time.
Dynamic RR	There is time to wait, whoever waits more takes it out.	It take less time.

✓ **The time improved in terms of Dynamic RR more than Static RR, because the Dynamic RR takes out those who wait more in time and takes it out first, while the static doesn't wait , but rather uses the priority of time.**

6. Reference

https://www.scientific.net/AMM.347-350.2203