**Program 4**

**Write an OpenMP program to convert a color image to black and white image. a) Demonstrate the performance of different scheduling techniques for varying chunk values**

**b) Analyze the scheduling patterns by assigning a single color value for an image for each thread**

#include <stdio.h>

#include <error.h>

#include <gd.h>

#include<string.h>

#include<omp.h>

int main(intargc, char \*\*argv)

{

intnt =4;

FILE \*fp,\*fp1 = {0};

gdImagePtrimg;

char \*iname = NULL;

char \*oname = NULL;

int color, x, y, w, h,i=0;

int red, green, blue,tmp,tid;

color = x = y = w = h = 0;

red = green = blue = 0;

if(argc != 3)

error(1, 0, "Usage: gdnegat input.png output.png");

else

{iname = argv[1];

oname = argv[2];

}if((fp = fopen(iname, "r")) == NULL)

error(1, 0, "Error - fopen(): %s", iname);

else

{img = gdImageCreateFromPng(fp);}

w = gdImageSX(img);

h = gdImageSY(img);

double t = omp\_get\_wtime();

omp\_set\_num\_threads(nt);

printf("width:%d \t height:%d\n",w,h);

#pragma omp parallel for private(y,color,red,blue,green) schedule(guided,50)/\*schedule(static) schedule(dynamic,50) schedule(guided,50)\*/

for(x = 0; x < w; x++)

{

for(y = 0; y < h; y++)

{

tid=omp\_get\_thread\_num();

color=gdImageGetPixel(img, x, y);

red = gdImageRed(img, color);

green = gdImageGreen(img, color);

blue =gdImageBlue(img, color);

tmp = (red+green+blue)/ 3;

red = green = blue = tmp;

if(tid==0)

{

color = gdImageColorAllocate(img, red, green, blue);

gdImageSetPixel(img, x, y, color);

}

if(tid==1)

{

color = gdImageColorAllocate(img, red, green, blue);

gdImageSetPixel(img, x, y, color);

}

if(tid==2)

{

color = gdImageColorAllocate(img, red, green, blue);

gdImageSetPixel(img, x, y, color);

}

if(tid==3)

{

color = gdImageColorAllocate(img, red,green, blue);

gdImageSetPixel(img, x, y, color);

}} }

t=omp\_get\_wtime()-t;

printf("\ntime taken : %lf threads : %d\n",t,nt);

fp1=fopen(oname,"w");

gdImagePng(img, fp1);

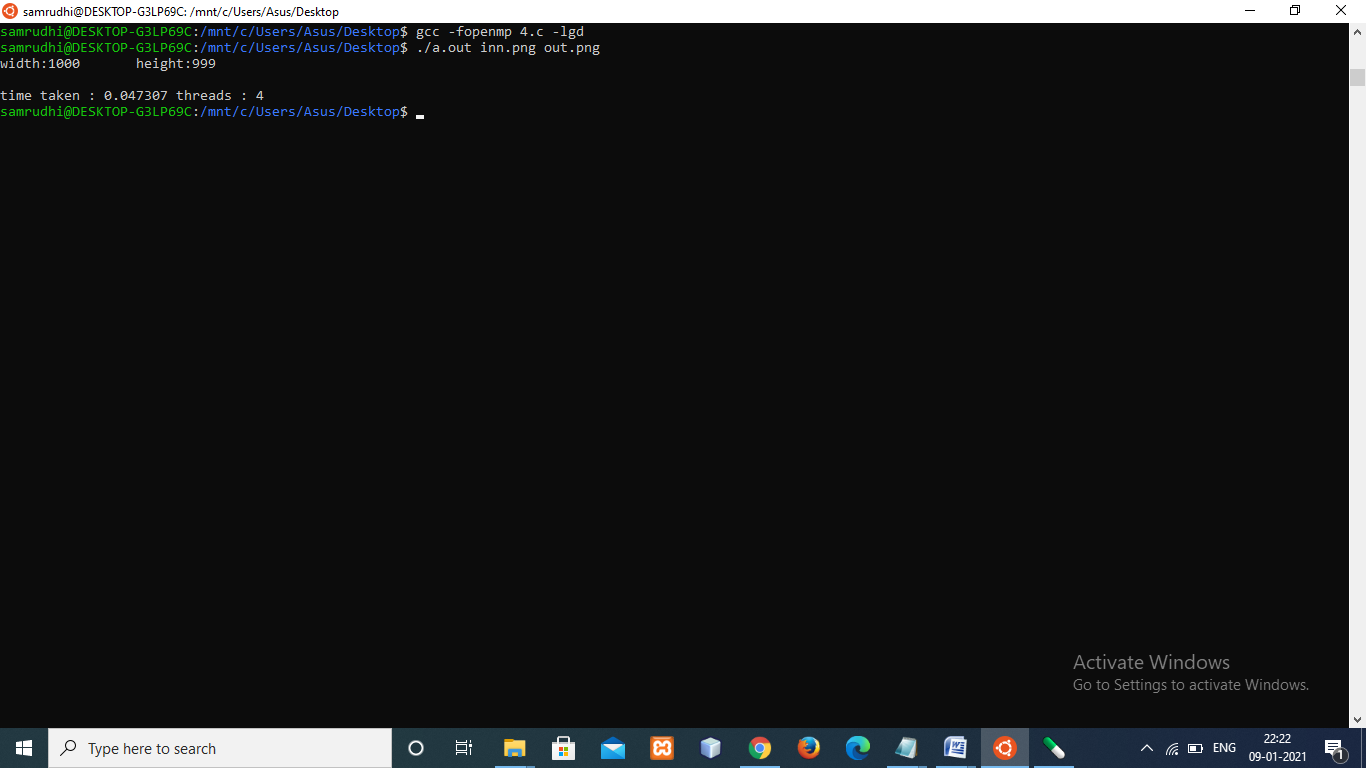
fclose(fp1);

gdImageDestroy(img);

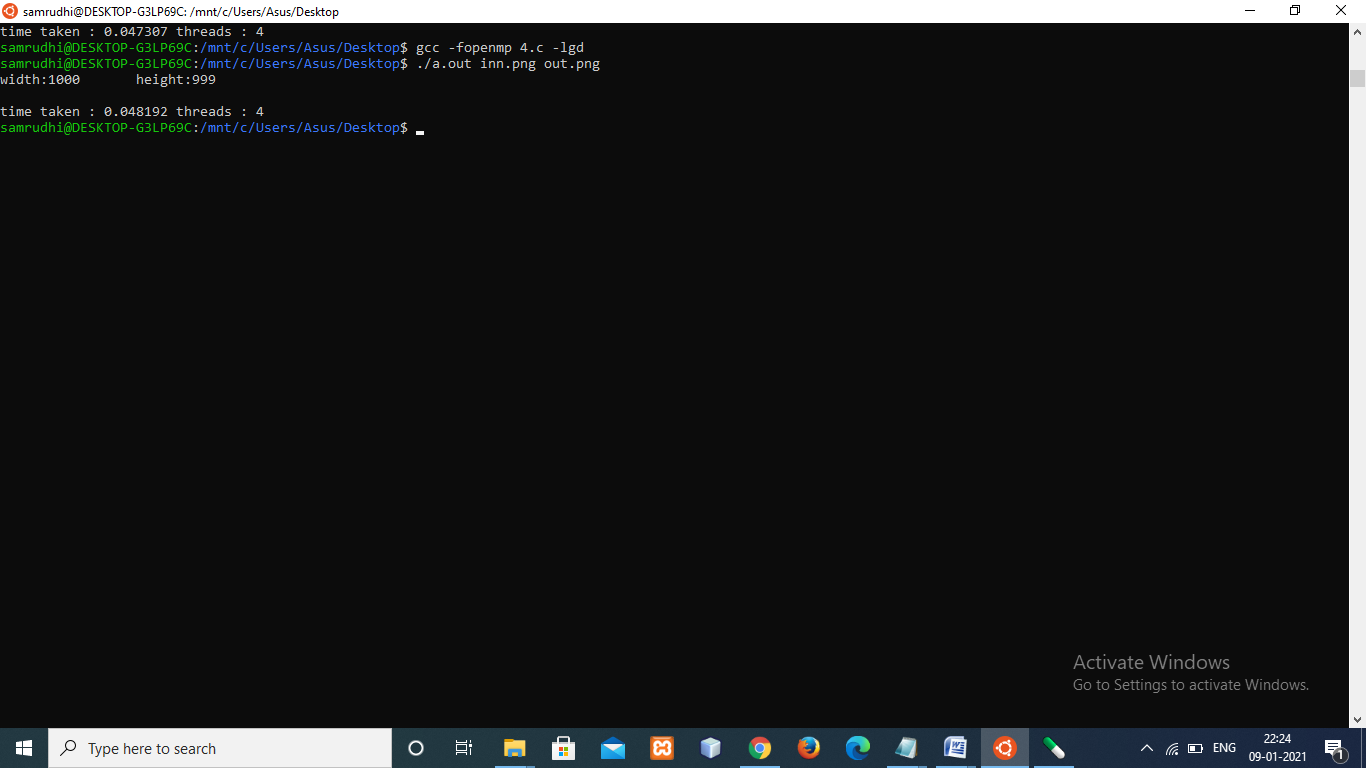
return 0;}

**output-**

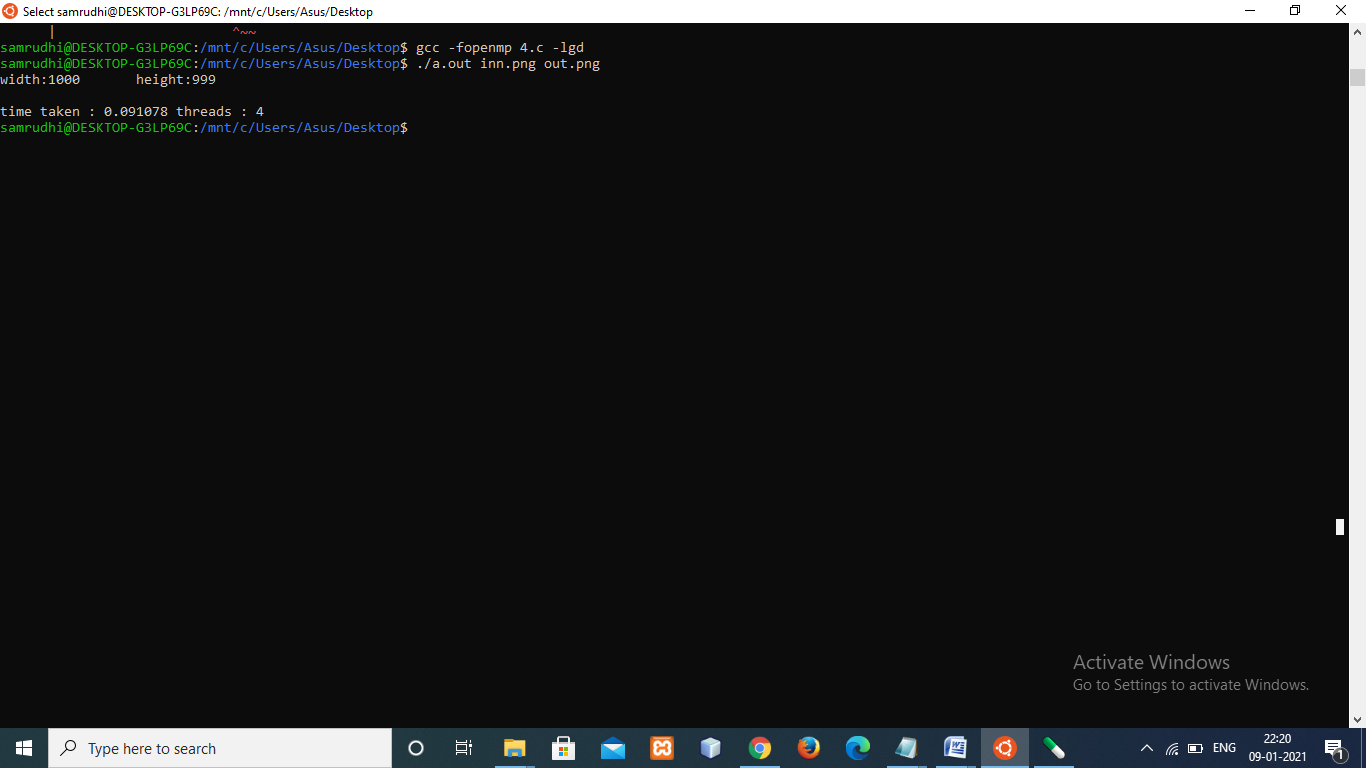
**1.schedule(static)**



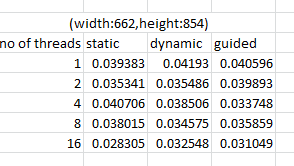
**2.schedule(dynamic,50)**



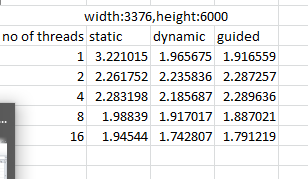
**3.schedule(guided,50)**



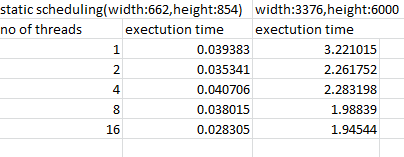
**Size of the image (width:662,height:854) with static, dynamic and guided Schedling**



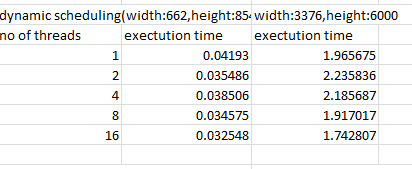
**Size of the image(width:3376,height:6000) with static, dynamic and Guided Scheduling**



**Static scheduling on different inputs and number of threads**



**Dynamic scheduling on different inputs and number of threads**



**Guided scheduling on different inputs and number of threads**

