Acer Cristea
Cruzid: acristea
Winter 2023 - CSE 13S
Dr. Veenstra

Assignment 1 DESIGN.pdf

Description of Program

Write a shell script, or more specifically, a bash script to produce plots of a Monte Carlo program written in C. The program prints the Monte Carlo estimation for π after each random point it tests. Monte Carlo estimation of π uses a large number of points in a square with an inscribed quadrant. The ratio of the number of points in the quadrant to the number of points in the square is the ratio of the two areas. We will uniformly scatter a number of points in a square, and measure the number of points that fall within the quadrant (distance from origin less than or equal to 1) and the other points. The C program will output the estimated value of π after every point for n number of points. We should end up with some plots that give us a visual representation of Monte Carlo's estimation.

The Deliverables
(files that need to be included in the directory "asgn1")

1. plot.sh: This bash script should produce the Monte Carlo method plots used in our report. This script should produce plots that give us a visual representation of Monte Carlo's estimation.

2. monte-carlo.c: This file is provided and contains the implementation of the Monte Carlo program.

3. Makefile: This file is provided and directs the compilation process of the Monte Carlo program.

4. README.md: This must use proper Markdown syntax. It must describe how to use your script and Makefile. should also list and explain any command-line options that your program accepts.

5. DESIGN.pdf: What I am currently writing, a description of the programs and the design process including supporting pseudocode.

6. WRITEUP.pdf: This document must be a proper PDF. This writeup must include the plots that you produced using your bash script, as well as discussion on which UNIX

commands you used to produce each plot and why you chose to use them. You should use LATEX to produce this document.

Pseudocode / Structure

*All the code you need to write will be in the bash file, plot.sh

Helpful Bash Documentation:	Bash Documentation:
	For loop → for n in a b c
	Do
	Echo $n
	Done
	
	Wc → finds out the statistics of a file (counts characters + lines)
	
	Sort → can sort a file lexicographically
	
	Head → returns the first ten lines, but if specified can return any number
	
	Uniq → returns the number of repetitions in the file
	
	>> and > → redirects code into files
	
	| → pipeline allows you to string multiple commands together and run it in one line
	
	Awk → allows you to manipulate what data should go into the file you are redirecting to
	
	If Statements → if [condition]; then
	…
	Else
	…
	fi

Step 1: Setting up bash	1. First specify the interpreter to be used, #!/bin/bash 2. Rebuild the Monte Carlo executable 3. Place the data points into a temporary file
Step 2: Creating the .dat file for the first graph	1. Use a loop statement to have the program generate a 1000 points inside the same temporary data file

	a. Use a for loop
Step 3: Plotting the first Monte Carlo Graph → *using palette* → *using gnuplot* → *using dat files*	1. Create a document that is sent to gnuplot (gnuplot <<END) a. Set the terminal as a pdf b. Name the file c. Give the graph a title d. Give the graph an x-label e. Give the graph an y-label f. Create a palette so that we can color the dots that are inside the circle and square or only inside the square i. We can unset this palette so it doesn't show up in the pdf g. Set an object - we want a circle - so users can have a visual representation of the circle h. Plot the data using our same temporary file 2. End the gnuplot.
Step 4: Creating the .dat file for the second graph	1. To create another graph we need different data. 2. Create another data file of n number of iterations with a unique seed 3. Grab the data from the column that has the estimated π and subtract that by π a. Can use awk for this 4. We can sort all of this into a new data file, which will have just a single column of data a. In the same awk line as above 5. This will generate 1 line of the graph 6. If we want n lines, we need to n number of times
Step 5: Plotting the second Monte Carlo Graph → *using gnuplot* → *using dat files*	1. Then to create another graph or plot, create another gnuplot document 2. Repeat similar steps from (Step 3) 3. Then plot each data file we just created a. This is because we are plotting multiple lines on the graph so we use multiple files 4. End the gnuplot document.