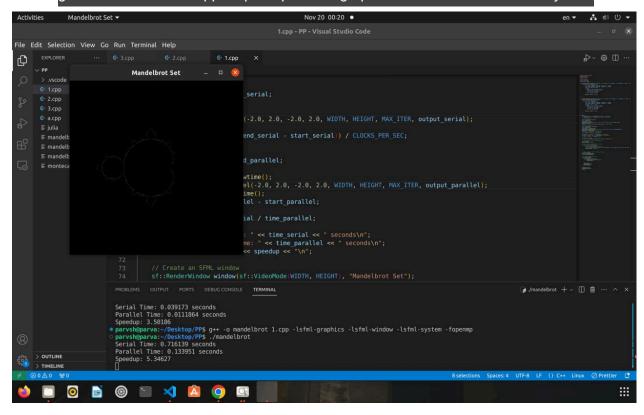
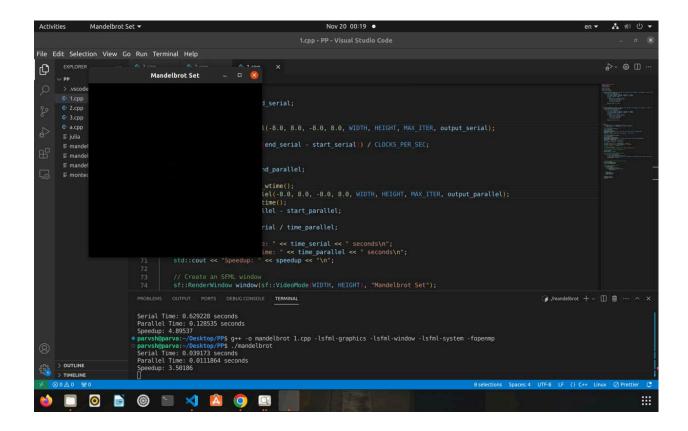
گلبو رشیدی 810100148 یروا شریفی 810100171

1. Mandelbrot set

g++ -o mandelbrot 1.cpp -fopenmp -lsfml-graphics -lsfml-window -lsfml-system



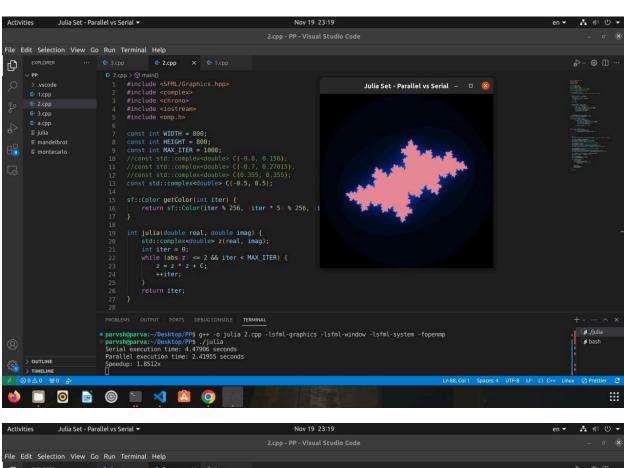


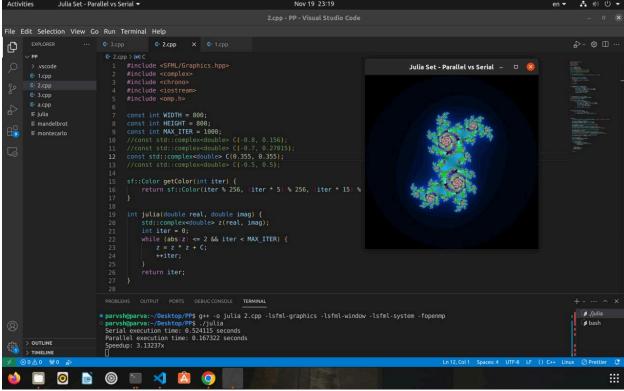
در روش سریال، برای هر پیکسل از تصویر، مختصات عددی آن را محاسبه کرده و تعداد تکرارهای لازم برای بررسی تعلق آن نقطه به مجموعه Mandelbrot را محاسبه میکنیم. این محاسبات بهطور خطی برای تمامی پیکسلها انجام میشود.

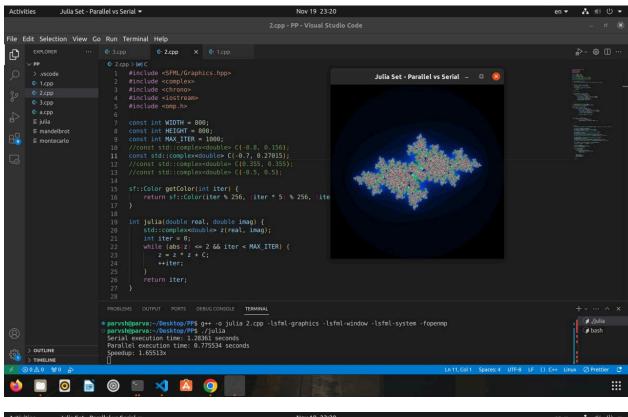
در روش موازی، دستور #pragma omp parallel for استفاده شده است تا حلقههای محاسباتی برای هر پیکسل بهطور همزمان در تردها اجرا شوند. از schedule(dynamic) برای اختصاص تسک به صورت دانامیک به تردها استفاده شده است که لود را بهطور متوازن تقسیم کند و هر ترد بخشی از محاسبات را انجام دهد. در داخل حلقههای محاسباتی، هر پیکسل بهطور مستقل از دیگر پیکسلها محاسبه میشود، بنابراین موازیسازی این محاسبات بسیار مؤثر است. از #pragma omp parallel for برای تقسیم حلقههای داخلی میان تردها استفاده شده است. این باعث میشود که محاسبات سریعتر انجام شوند، زیرا بهطور همزمان به محاسبه بخشهای مختلف تصویر پرداخته میشود.

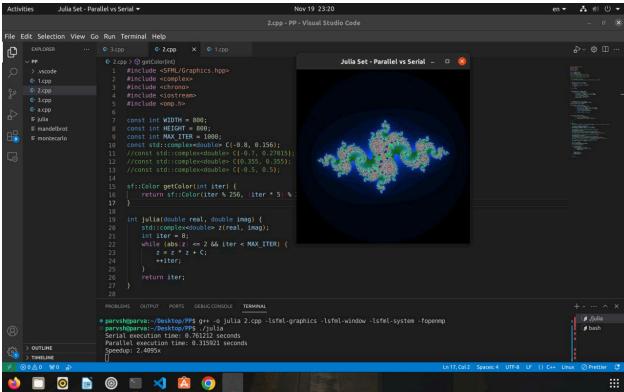
2. Julia Set

g++ -o julia 2.cpp -lsfml-graphics -lsfml-window -lsfml-system -fopenmp







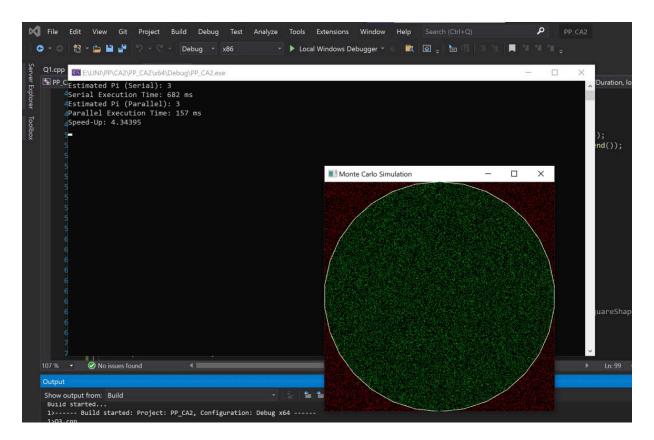


تابع computeSerial () مجموعه Julia را برای هر پیکسل در تصویر بهطور سریال محاسبه میکند. مختصات حقیقی و موهومی هر پیکسل از موقعیت آن در تصویر محاسبه و سیس برای هر نقطه تابع julia () فراخوانی میشود. رنگ هر پیکسل بر اساس تعداد تکرارها در تابع julia () تعیین میشود.

تابع computeParallel () همان عمليات را بهطور موازی انجام میدهد:

دستور #pragma omp parallel num_threads(8#) به کار رفته است تا پردازشها به ۸ ترد تقسیم شوند.برای اجتناب از تداخل در نوشتن دادهها، ابتدا رنگها در یک بافر محلی ذخیره میشوند.

3. Monte Carlo



حالت سریال (performPiCalcSerial):

در این حالت، تمام نقاط بهطور خطی پردازش میشوند.نقاط داخل دایره با رنگ سبز و نقاط خارج از دایره با رنگ قرمز نمایش داده میشوند. برای هر نقطه، بررسی میشود که آیا در داخل دایره است یا خیر، و سپس شمارش انجام میشود. حالت موازی (performPiCalcParallel):

در این حالت، نقاط تصادفی بهطور موازی بین تردها تقسیم میشوند. برای جلوگیری از مشکلات همزمانی، از thread_local برای تولید اعداد تصادفی منحصر به فرد برای هر ترد استفاده شده است. پس از محاسبه تعداد نقاط داخل و خارج دایره در هر ترد، نتایج با استفاده از #pragma omp critical بهروزرسانی میشوند.