بهاره پورداد / محمدرضا عظیمی

810196510 /810896045

«تمرین سوم برنامه نویسی موازی» **تاریخ تحویل: 4 آذر**

سوال1:

ابتدا کتابخانه های مورد نیاز را تعریف میکنیم.

سپس متغیرهای مربوط به اندازه گیری زمان(طول clock cycle) ها را تعریف میکنیم.

هدف این سوال این است که:

دو تصویر را بخوانیم، سپس با استفاده از data structure مربوط به تصاویر، به ماتریسی که تمام پیکسل هارا بصورت عدد ذخیره کرده دسترسی پیدا کنیم تا با محاسبه ی نظیر به نظیر درایه ها و انجام عمل قدرمطلق تفاضل روی آنها، بفهمیم که تصویر متحرک است. (یعنی که دو تصویر در جزئیات باهم تفاوت دارند).

این کارها را باید با دو پیاده سازی serial و parallel انجام دهیم تا با محاسبه ی زمان اجرا، میزان تسریع برنامه سریال نسبت به موازی را تخمین بزنیم.

- ابتدا با استفاده از تابع آماده مربوط به opencv که همان imread است تصاویر را میخوانیم.
- هدف ما دسترسی به ماتریس پیکسل هاست. در نتیجه دو اشاره گر برای ابتدای ماتریس دو تصویر تعریف میکنیم. (باید توجه داشت که نوع متغیر با نوع داده های ماتریس متفاوت است پس cast میکنیم). علاوه بر این یک ماتریس هم برای قرار دادن خروجی هایی که محاسبه خواهیم کرد، درنظر میگیریم.
- ابتدا بصورت سریال پیاده سازی میکنیم. میدانیم که Hotspotهای برنامه ما در مرحله محاسبه است. در نتیجه از این نقطه شروع به زمان گیری میکینم. برای خواندن ماتریس ها به دو حلقه تودرتو احتیاج داریم. بصورت نظیر به نظیر قدرمطلق تفاضل دو ماتریس را محاسبه کرده و در ماتریس خروجی که در ابتدا تعریف کردیم میریزیم. زمان گیری را خاتمه میدهیم و مدت زمان اجرا را در time1 میریزیم.
- حالا باید برنامه را بصورت موازی پیاده سازی کنیم. تعاریف مربوط را با کمک SSE انجام میدهیم. مجددا cast های لازم را انجام میدهیم. زمان گیری را برای بخش محاسبه آغاز میکنیم. حلقه های تودرتو را برای دسترسی به ماتریس در نظر میگیریم. دو ماتریس را به ترتیب در m1 و m2 لود میکنیم. برای محاسبه تفاضل باید یکبار a – b و بار دیگر b – a را درنظر بگیریم و نتیجه را OR کنیم. نتیجه را درون m5 میریزیم و درایه ها را در ماتریس پیکسل های خروجی ذخیره میکینم. زمان گیری را خاتمه میدهیم و طول زمان صرف شده را در time2 میریزیم.
- چون در این سوال نیازی به نمایش تصویر خروجی نداشتیم و فقط هدف محاسبه زمان اجرا بود، در نهایت زمان های محاسبه شده برای هر دو پیاده سازی را نمایش داده و با تقسیم زمان سریال به موازی، Speedup یا میزان تسریع را نمایش میدهیم.

سوال2:

ابتدا کتابخانه های مورد نیاز را تعریف میکنیم.

سپس متغیرهای مربوط به اندازه گیری زمان(طول clock cycle) ها را تعریف میکنیم.

هدف این سوال این است که:

دو تصویر را بخوانیم، تصویر دوم را با فرمول مربوط شفاف کرده و به تصویر اول اضافه کنیم و زمان اجرای دو پیاده سازی سریال و موازی را اندازه گرفته و میزان تسریع برنامه در حالت موازی را محاسبه کنیم.

- ابتدا با استفاده از تابع آماده مربوط به opencv که همان imread است تصاویر را میخوانیم.

- هدف ما دسترسی به ماتریس پیکسل هاست. در نتیجه دو اشاره گر برای ابتدای ماتریس دو تصویر تعریف میکنیم. (باید توجه داشت که نوع متغیر با نوع داده های ماتریس متفاوت است پس cast میکنیم). علاوه بر این یک ماتریس هم برای قرار دادن خروجی هایی که محاسبه خواهیم کرد، درنظر میگیریم.

- ابتدا بصورت سریال پیاده سازی میکنیم. میدانیم که Hotspotهای برنامه ما در مرحله محاسبه است. در نتیجه از این نقطه شروع به زمان گیری میکینم. برای خواندن ماتریس ها به دو حلقه تودرتو احتیاج داریم. بصورت نظیر به نظیر ابتدا ثابت alpha را در درایه های ماتریس دوم ضرب کرده و حاصل را با درایه های ماتریس اول جمع میکنیم. زمان گیری را خاتمه میدهیم و مدت زمان اجرا را در time1 میریزیم.

- حالا باید برنامه را بصورت موازی پیاده سازی کنیم. تعاریف مربوط را با کمک SSE انجام میدهیم. مجددا cast های لازم را انجام میدهیم. زمان گیری را برای بخش محاسبه آغاز میکنیم. حلقه های تودرتو را برای دسترسی به ماتریس در نظر میگیریم. دو ماتریس را به ترتیب در m1 و m2 لود میکنیم.با استفاده از دستورات مربوط به SIMD عناصر ماتریس اول را با حاصلضرب عناصر ماتریس دوم در alpha جمع میکنیم. نتیجه را درون m3 میریزیم و درایه ها را در ماتریس پیکسل های خروجی ذخیره میکینم. زمان گیری را خاتمه میدهیم و طول زمان صرف شده را در time2 میریزیم.

- در این سوال چون باید تصویر خروجی را نمایش دهیم، ابتدا با استفاده از تابع namedWindow یک پنجره تعریف میکنیم و با استفاده از تابع imshow تصویر خروجی را درون پنجره ای که تعریف کردیم نمایش میدهیم. این توابع مربوط به کتابخانه opencv هستند. علاوه بر تصویر باید زمان های محاسبه شده در دو پیاده سازی و میزان تسریع را هم نمایش دهیم.

«پایان»