به‌نام خدا

۱. در این بخش برای موازی‌سازی از ساختار for درون ساختار parallel استفاده می‌کنیم. ابتدا تمامی متغیرها را به‌صورت پیش‌فرض با default(shared) مشترک در نظر می‌گیریم. سپس از دو متغیر محلی localMax و localMaxIdx برای هر ترد استفاده می‌کنیم تا هر ترد با دیتای خودش ماکسیمم را حساب کند و نهایتاً برای این‌که تردها بتوانند مقدار ماکسیمم کلی را تغییر بدهند از ساختار critical استفاده می‌کنیم تا عنصر ماکسیمم و اندیس مربوط به آن را بدون مشکل همگام‌سازی، به‌روز کنیم:

```
#pragma omp parallel default(shared) private(localMax, localMaxIdx) num_threads(4)
  {
    localMax = fArray[0];
    localMaxIdx = 0;
    #pragma omp for
        for (idx = 0; idx < VECTOR_SIZE; idx++)
        {
          localMax = localMax > fArray[idx] ? localMax : fArray[idx];
          localMaxIdx = localMax > fArray[idx] ? localMaxIdx : idx;
        }
    #pragma omp critical
    {
      if (localMax > maxVal)
      {
        maxVal = localMax;
        maxIdx = localMaxIdx;
      }
    }
  }
```

پس از اجرا مشاهده می‌کنیم که پاسخ‌ها با هم برابر هستند و هم‌چنین تسریعی که به دست آمده ۲.۰۳ برابر است.

```
~/Courses/9-Fall99/PP/CAs/CA4/1 ./main
Hossein Soltanloo
Student Number: 810195407
The serial result is  Value = 100.000000    Index = 1310
The parallel result is  Value = 100.000000    Index = 1310
Serial Run time = 5381259
Parallel Run time = 2642384
Speedup = 2.036517
```

۲. در این قسمت ابتدا به مشاهده‌ی نتایج می‌پردازیم:

```
~/Courses/9-Fall99/PP/CAs/CA4/2 ./main
Hossein Soltanloo
Student Number: 810195407

Serial Run time = 416236675
Parallel (section) Run time = 1094325585
Parallel (task) Run time = 167133938
Parallel (section) Speedup = 0.380359
Parallel (task) Speedup = 2.490438

Test if arrays are sorted:
Serial:
Array sorted.
Parallel (section):
Array sorted.
Parallel (task):
Array sorted.
```

اولین موردی که به آن باید اشاره شود این است که در quicksort آرایه با pivot به دو بخش بزرگ‌تر و کوچک‌تر از pivot تقسیم می‌شود که به‌طور بازگشتی این بخش‌ها خودشان شکسته می‌شوند تا به تک عنصر برسیم و این‌گونه عمل مرتب‌سازی انجام می‌گیرد. حال برای انجام این کار با OpenMP، اولین چیزی که به ذهن می‌رسد این است که این دو بخش‌ها را بین دو section تقسیم کنیم که به همین ترتیب به‌صورت بازگشتی، sectionهای دیگر درون این‌ها ایجاد می‌شوند و در نتیجه تعداد زیادی ترد ایجاد می‌شود. حال وقتی به میزان تسریع نگاه می‌کنیم می‌بینیم که با این کار، عمل مرتب‌سازی بسیار کندتر شده است که این ناشی از این موضوع است که در هر مرحله تعداد بسیار زیادی ترد ساخته می‌شود و از بین می‌رود که همین کار سربار بسیار زیادی دارد که نهایتاً آن را از حالت سریال هم کندتر می‌کند. در نتیجه از ویژگی جدیدتری که OpenMP در اختیارمان قرار داده است استفاده می‌کنیم تا این مشکل را حل کنیم. این ویژگی جدید Task نام دارد که مخصوص همین مواقعی ایجاد شده است که مسئله‌مان بازگشتی است و از قبل میزان تردها مشخص نیست. در این حالت یک استخر از taskها ایجاد می‌شود و در طرف دیگر تعداد مشخصی ترد وجود دارد. وقتی که به ساختار task می‌رسیم، به ازای آن در استخر یک تسک ایجاد می‌شود و تردهایی که وجود دارند مسئول اجرای تسک‌های درون این استخر یا pool خواهند بود. بدین ترتیب از ساختن بی‌رویه‌ی تردها جلوگیری می‌شود و سربار بسیار کاهش پیدا می‌کند تا جایی که در تصویر مشاهده می‌شود میزان تسریع ۲.۴۹ برابر است. نهایتاً مشاهده می‌شود که در همه‌ی حالت‌ها آرایه به‌درستی مرتب شده است.

۳. زمان اجرای برنامه‌ی سریال:

```
~/Courses/9-Fall99/PP/CAs/CA4/3/Serial ./main
Hossein Soltanloo - 810195407

Serial timing for 100000 iterations

Time Elapsed      26195 mSecs Total=32.617277 Check Sum = 100000
```

نکته: جهت افزودن زمان‌های اجرای هر ترد، قید nowait افزوده شده است تا زمان اجرای خالص هر ترد به دست آید.

- زمان اجرای برنامه‌ی موازی در حالت static:

(میانگین: ۷۹۲۶ میلی‌ثانیه، میزان تسریع: ۳.۳)

```
~/Courses/9-Fall99/PP/CAs/CA4/3/Parallel set OMP_SCHEDULE="static"
~/Courses/9-Fall99/PP/CAs/CA4/3/Parallel ./main
Hossein Soltanloo - 810195407

OpenMP Parallel Timings for 100000 iterations

Time Elapsed       7531 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7530 mSecs
T1 Time Elapsed       7530 mSecs
T2 Time Elapsed       7530 mSecs
T3 Time Elapsed       7530 mSecs
Time Elapsed       7840 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7840 mSecs
T1 Time Elapsed       7840 mSecs
T2 Time Elapsed       7840 mSecs
T3 Time Elapsed       7839 mSecs
Time Elapsed       7905 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7905 mSecs
T1 Time Elapsed       7905 mSecs
T2 Time Elapsed       7905 mSecs
T3 Time Elapsed       7905 mSecs
Time Elapsed       8043 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       8043 mSecs
T1 Time Elapsed       8043 mSecs
T2 Time Elapsed       8043 mSecs
T3 Time Elapsed       8043 mSecs
Time Elapsed       8394 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       8393 mSecs
T1 Time Elapsed       8394 mSecs
T2 Time Elapsed       8393 mSecs
T3 Time Elapsed       8393 mSecs
Time Elapsed       7843 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7842 mSecs
T1 Time Elapsed       7843 mSecs
T2 Time Elapsed       7843 mSecs
T3 Time Elapsed       7842 mSecs
```

- زمان اجرای برنامه‌ی موازی در حالت dynamic, 1000:

(میانگین: ۷۴۹۹ میلی‌ثانیه، میزان تسریع: ۳.۴۹)

```
~/Courses/9-Fall99/PP/CAs/CA4/3/Parallel set OMP_SCHEDULE="dynamic, 1000"
~/Courses/9-Fall99/PP/CAs/CA4/3/Parallel ./main
Hossein Soltanloo - 810195407

OpenMP Parallel Timings for 100000 iterations

Time Elapsed       7297 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7296 mSecs
T1 Time Elapsed       7296 mSecs
T2 Time Elapsed       7296 mSecs
T3 Time Elapsed       7296 mSecs
Time Elapsed       7324 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7324 mSecs
T1 Time Elapsed       7324 mSecs
T2 Time Elapsed       7324 mSecs
T3 Time Elapsed       7324 mSecs
Time Elapsed       7386 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7386 mSecs
T1 Time Elapsed       7386 mSecs
T2 Time Elapsed       7386 mSecs
T3 Time Elapsed       7386 mSecs
Time Elapsed       7570 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7570 mSecs
T1 Time Elapsed       7570 mSecs
T2 Time Elapsed       7570 mSecs
T3 Time Elapsed       7570 mSecs
Time Elapsed       7882 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7882 mSecs
T1 Time Elapsed       7882 mSecs
T2 Time Elapsed       7882 mSecs
T3 Time Elapsed       7882 mSecs
Time Elapsed       7535 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7535 mSecs
T1 Time Elapsed       7535 mSecs
T2 Time Elapsed       7535 mSecs
T3 Time Elapsed       7535 mSecs
```

- زمان اجرای برنامه‌ی موازی در حالت dynamic, 2000:

(میانگین: ۷۵۴۱ میلی‌ثانیه، میزان تسریع: ۳.۴۷)

```
~/Courses/9-Fall99/PP/CAs/CA4/3/Parallel set OMP_SCHEDULE="dynamic, 2000"
~/Courses/9-Fall99/PP/CAs/CA4/3/Parallel ./main
Hossein Soltanloo - 810195407

OpenMP Parallel Timings for 100000 iterations

Time Elapsed       7039 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7038 mSecs
T1 Time Elapsed       7038 mSecs
T2 Time Elapsed       7038 mSecs
T3 Time Elapsed       7038 mSecs
Time Elapsed       7420 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7420 mSecs
T1 Time Elapsed       7420 mSecs
T2 Time Elapsed       7420 mSecs
T3 Time Elapsed       7420 mSecs
Time Elapsed       7794 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7793 mSecs
T1 Time Elapsed       7794 mSecs
T2 Time Elapsed       7794 mSecs
T3 Time Elapsed       7794 mSecs
Time Elapsed       7543 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7543 mSecs
T1 Time Elapsed       7542 mSecs
T2 Time Elapsed       7542 mSecs
T3 Time Elapsed       7543 mSecs
Time Elapsed       7812 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7812 mSecs
T1 Time Elapsed       7812 mSecs
T2 Time Elapsed       7812 mSecs
T3 Time Elapsed       7812 mSecs
Time Elapsed       7641 mSecs Total=32.617277 Check Sum = 100000
T0 Time Elapsed       7640 mSecs
T1 Time Elapsed       7641 mSecs
T2 Time Elapsed       7640 mSecs
T3 Time Elapsed       7640 mSecs
```

مشاهده می‌شود که در همه‌ی حالت‌ها با توجه به زمان اجرای ریسمان‌ها، بار توزیع‌شده کاملاً بین آن‌ها متوازن است.