Ex 5.9 page 264

Code:

```
/* File:    omp_trap3.c
 * Prabhat Bhootra
 */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <omp.h>

void Usage(char* prog_name);
double f(double x);    /* Function we're integrating */
double Trap(double a, double b, int n, int thread_count, int* iterationsArray);

int main(int argc, char* argv[]) {
  double  global_result = 0.0;  /* Store result in global_result */
  double  a, b;               /* Left and right endpoints      */
  int     n;                  /* Total number of trapezoids    */
  int        z, y;
  int     thread_count;
  if (argc != 2) Usage(argv[0]);
  thread_count = strtol(argv[1], NULL, 10);
  printf("Enter a, b, and n\n");
  scanf("%lf %lf %d", &a, &b, &n);
  int* iterations = malloc(sizeof(int)*n);
  global_result = Trap(a, b, n, thread_count, iterations);
  printf("With n = %d trapezoids, our estimate\n", n);
  printf("of the integral from %f to %f = %.14e\n",
    a, b, global_result);
```

```
  for (z = 1; z < n; z++) {
    printf("%d iteration of the for loop is assigned to thread %d\n", z, iterations[z]);
  }
  return 0;
}  /* main */

void Usage(char* prog_name) {
  fprintf(stderr, "usage: %s <number of threads>\n", prog_name);
  exit(0);
}  /* Usage */

double f(double x) {
  double return_val;
  return_val = x*x;
  return return_val;
}  /* f */

double Trap(double a, double b, int n, int thread_count, int* iterationsArray) {
  double  h, approx;
  int  i;
  h = (b-a)/n;
  approx = (f(a) + f(b))/2.0;
#  pragma omp parallel for num_threads(thread_count) \
    reduction(+: approx) schedule(runtime)
  for (i = 1; i <= n-1; i++) {
   approx += f(a + i*h);
   iterationsArray[i] = omp_get_thread_num();
  }
  approx = h*approx;
```

```
  return approx;
} /* Trap */
```

Test results:

For all test runs, a = 1, b = 100, n = 50 and number of threads = 16

1. **Default scheduling:**

   Enter a, b, and n

   1 100 50

   With n = 50 trapezoids, our estimate

   of the integral from 1.000000 to 100.000000 = 3.33397686600000e+05

   1 iteration of the for loop is assigned to thread 0

   2 iteration of the for loop is assigned to thread 0

   3 iteration of the for loop is assigned to thread 0

   4 iteration of the for loop is assigned to thread 0

   5 iteration of the for loop is assigned to thread 1

   6 iteration of the for loop is assigned to thread 1

   7 iteration of the for loop is assigned to thread 1

   8 iteration of the for loop is assigned to thread 2

   9 iteration of the for loop is assigned to thread 2

   10 iteration of the for loop is assigned to thread 2

   11 iteration of the for loop is assigned to thread 3

   12 iteration of the for loop is assigned to thread 3

   13 iteration of the for loop is assigned to thread 3

   14 iteration of the for loop is assigned to thread 4

   15 iteration of the for loop is assigned to thread 4

   16 iteration of the for loop is assigned to thread 4

   17 iteration of the for loop is assigned to thread 5

   18 iteration of the for loop is assigned to thread 5

   19 iteration of the for loop is assigned to thread 5

20 iteration of the for loop is assigned to thread 6

21 iteration of the for loop is assigned to thread 6

22 iteration of the for loop is assigned to thread 6

23 iteration of the for loop is assigned to thread 7

24 iteration of the for loop is assigned to thread 7

25 iteration of the for loop is assigned to thread 7

26 iteration of the for loop is assigned to thread 8

27 iteration of the for loop is assigned to thread 8

28 iteration of the for loop is assigned to thread 8

29 iteration of the for loop is assigned to thread 9

30 iteration of the for loop is assigned to thread 9

31 iteration of the for loop is assigned to thread 9

32 iteration of the for loop is assigned to thread 10

33 iteration of the for loop is assigned to thread 10

34 iteration of the for loop is assigned to thread 10

35 iteration of the for loop is assigned to thread 11

36 iteration of the for loop is assigned to thread 11

37 iteration of the for loop is assigned to thread 11

38 iteration of the for loop is assigned to thread 12

39 iteration of the for loop is assigned to thread 12

40 iteration of the for loop is assigned to thread 12

41 iteration of the for loop is assigned to thread 13

42 iteration of the for loop is assigned to thread 13

43 iteration of the for loop is assigned to thread 13

44 iteration of the for loop is assigned to thread 14

45 iteration of the for loop is assigned to thread 14

46 iteration of the for loop is assigned to thread 14

47 iteration of the for loop is assigned to thread 15

48 iteration of the for loop is assigned to thread 15

49 iteration of the for loop is assigned to thread 15

2. **Static scheduling with chunk size = 1:**

Enter a, b, and n
1 100 50
With n = 50 trapezoids, our estimate
of the integral from 1.000000 to 100.000000 = 3.33397686600000e+05
1 iteration of the for loop is assigned to thread 0
2 iteration of the for loop is assigned to thread 1
3 iteration of the for loop is assigned to thread 2
4 iteration of the for loop is assigned to thread 3
5 iteration of the for loop is assigned to thread 4
6 iteration of the for loop is assigned to thread 5
7 iteration of the for loop is assigned to thread 6
8 iteration of the for loop is assigned to thread 7
9 iteration of the for loop is assigned to thread 8
10 iteration of the for loop is assigned to thread 9
11 iteration of the for loop is assigned to thread 10
12 iteration of the for loop is assigned to thread 11
13 iteration of the for loop is assigned to thread 12
14 iteration of the for loop is assigned to thread 13
15 iteration of the for loop is assigned to thread 14
16 iteration of the for loop is assigned to thread 15
17 iteration of the for loop is assigned to thread 0
18 iteration of the for loop is assigned to thread 1
19 iteration of the for loop is assigned to thread 2
20 iteration of the for loop is assigned to thread 3
21 iteration of the for loop is assigned to thread 4
22 iteration of the for loop is assigned to thread 5
23 iteration of the for loop is assigned to thread 6
24 iteration of the for loop is assigned to thread 7
25 iteration of the for loop is assigned to thread 8
26 iteration of the for loop is assigned to thread 9
27 iteration of the for loop is assigned to thread 10
28 iteration of the for loop is assigned to thread 11
29 iteration of the for loop is assigned to thread 12
30 iteration of the for loop is assigned to thread 13
31 iteration of the for loop is assigned to thread 14
32 iteration of the for loop is assigned to thread 15
33 iteration of the for loop is assigned to thread 0
34 iteration of the for loop is assigned to thread 1
35 iteration of the for loop is assigned to thread 2
36 iteration of the for loop is assigned to thread 3
37 iteration of the for loop is assigned to thread 4
38 iteration of the for loop is assigned to thread 5

39 iteration of the for loop is assigned to thread 6
40 iteration of the for loop is assigned to thread 7
41 iteration of the for loop is assigned to thread 8
42 iteration of the for loop is assigned to thread 9
43 iteration of the for loop is assigned to thread 10
44 iteration of the for loop is assigned to thread 11
45 iteration of the for loop is assigned to thread 12
46 iteration of the for loop is assigned to thread 13
47 iteration of the for loop is assigned to thread 14
48 iteration of the for loop is assigned to thread 15
49 iteration of the for loop is assigned to thread 0

3. **Static scheduling with chunk size 2:**

Enter a, b, and n
1 100 50
With n = 50 trapezoids, our estimate
of the integral from 1.000000 to 100.000000 = 3.33397686600000e+05
1 iteration of the for loop is assigned to thread 0
2 iteration of the for loop is assigned to thread 0
3 iteration of the for loop is assigned to thread 1
4 iteration of the for loop is assigned to thread 1
5 iteration of the for loop is assigned to thread 2
6 iteration of the for loop is assigned to thread 2
7 iteration of the for loop is assigned to thread 3
8 iteration of the for loop is assigned to thread 3
9 iteration of the for loop is assigned to thread 4
10 iteration of the for loop is assigned to thread 4
11 iteration of the for loop is assigned to thread 5
12 iteration of the for loop is assigned to thread 5
13 iteration of the for loop is assigned to thread 6
14 iteration of the for loop is assigned to thread 6
15 iteration of the for loop is assigned to thread 7
16 iteration of the for loop is assigned to thread 7
17 iteration of the for loop is assigned to thread 8
18 iteration of the for loop is assigned to thread 8
19 iteration of the for loop is assigned to thread 9
20 iteration of the for loop is assigned to thread 9
21 iteration of the for loop is assigned to thread 10
22 iteration of the for loop is assigned to thread 10
23 iteration of the for loop is assigned to thread 11
24 iteration of the for loop is assigned to thread 11
25 iteration of the for loop is assigned to thread 12
26 iteration of the for loop is assigned to thread 12
27 iteration of the for loop is assigned to thread 13
28 iteration of the for loop is assigned to thread 13

29 iteration of the for loop is assigned to thread 14
30 iteration of the for loop is assigned to thread 14
31 iteration of the for loop is assigned to thread 15
32 iteration of the for loop is assigned to thread 15
33 iteration of the for loop is assigned to thread 0
34 iteration of the for loop is assigned to thread 0
35 iteration of the for loop is assigned to thread 1
36 iteration of the for loop is assigned to thread 1
37 iteration of the for loop is assigned to thread 2
38 iteration of the for loop is assigned to thread 2
39 iteration of the for loop is assigned to thread 3
40 iteration of the for loop is assigned to thread 3
41 iteration of the for loop is assigned to thread 4
42 iteration of the for loop is assigned to thread 4
43 iteration of the for loop is assigned to thread 5
44 iteration of the for loop is assigned to thread 5
45 iteration of the for loop is assigned to thread 6
46 iteration of the for loop is assigned to thread 6
47 iteration of the for loop is assigned to thread 7
48 iteration of the for loop is assigned to thread 7
49 iteration of the for loop is assigned to thread 8

4. **Dynamic scheduling no chunk size specified (chunk size = 1):**

Enter a, b, and n
1 100 50
With n = 50 trapezoids, our estimate
of the integral from 1.000000 to 100.000000 = 3.33397686600000e+05
1 iteration of the for loop is assigned to thread 5
2 iteration of the for loop is assigned to thread 3
3 iteration of the for loop is assigned to thread 4
4 iteration of the for loop is assigned to thread 0
5 iteration of the for loop is assigned to thread 3
6 iteration of the for loop is assigned to thread 0
7 iteration of the for loop is assigned to thread 3
8 iteration of the for loop is assigned to thread 0
9 iteration of the for loop is assigned to thread 3
10 iteration of the for loop is assigned to thread 0
11 iteration of the for loop is assigned to thread 3
12 iteration of the for loop is assigned to thread 0
13 iteration of the for loop is assigned to thread 3
14 iteration of the for loop is assigned to thread 0
15 iteration of the for loop is assigned to thread 3
16 iteration of the for loop is assigned to thread 0
17 iteration of the for loop is assigned to thread 3
18 iteration of the for loop is assigned to thread 0

19 iteration of the for loop is assigned to thread 3
20 iteration of the for loop is assigned to thread 3
21 iteration of the for loop is assigned to thread 0
22 iteration of the for loop is assigned to thread 3
23 iteration of the for loop is assigned to thread 0
24 iteration of the for loop is assigned to thread 3
25 iteration of the for loop is assigned to thread 0
26 iteration of the for loop is assigned to thread 3
27 iteration of the for loop is assigned to thread 3
28 iteration of the for loop is assigned to thread 0
29 iteration of the for loop is assigned to thread 3
30 iteration of the for loop is assigned to thread 0
31 iteration of the for loop is assigned to thread 3
32 iteration of the for loop is assigned to thread 0
33 iteration of the for loop is assigned to thread 3
34 iteration of the for loop is assigned to thread 0
35 iteration of the for loop is assigned to thread 3
36 iteration of the for loop is assigned to thread 0
37 iteration of the for loop is assigned to thread 3
38 iteration of the for loop is assigned to thread 0
39 iteration of the for loop is assigned to thread 3
40 iteration of the for loop is assigned to thread 0
41 iteration of the for loop is assigned to thread 3
42 iteration of the for loop is assigned to thread 0
43 iteration of the for loop is assigned to thread 3
44 iteration of the for loop is assigned to thread 0
45 iteration of the for loop is assigned to thread 3
46 iteration of the for loop is assigned to thread 3
47 iteration of the for loop is assigned to thread 0
48 iteration of the for loop is assigned to thread 3
49 iteration of the for loop is assigned to thread 0

5. **Dynamic Scheduling with chunk size = 2:**

Enter a, b, and n
1 100 50
With n = 50 trapezoids, our estimate
of the integral from 1.000000 to 100.000000 = 3.33397686600000e+05
1 iteration of the for loop is assigned to thread 4
2 iteration of the for loop is assigned to thread 4
3 iteration of the for loop is assigned to thread 5
4 iteration of the for loop is assigned to thread 5
5 iteration of the for loop is assigned to thread 5
6 iteration of the for loop is assigned to thread 5
7 iteration of the for loop is assigned to thread 5
8 iteration of the for loop is assigned to thread 5

9 iteration of the for loop is assigned to thread 5
10 iteration of the for loop is assigned to thread 5
11 iteration of the for loop is assigned to thread 5
12 iteration of the for loop is assigned to thread 5
13 iteration of the for loop is assigned to thread 5
14 iteration of the for loop is assigned to thread 5
15 iteration of the for loop is assigned to thread 5
16 iteration of the for loop is assigned to thread 5
17 iteration of the for loop is assigned to thread 4
18 iteration of the for loop is assigned to thread 4
19 iteration of the for loop is assigned to thread 5
20 iteration of the for loop is assigned to thread 5
21 iteration of the for loop is assigned to thread 4
22 iteration of the for loop is assigned to thread 4
23 iteration of the for loop is assigned to thread 5
24 iteration of the for loop is assigned to thread 5
25 iteration of the for loop is assigned to thread 4
26 iteration of the for loop is assigned to thread 4
27 iteration of the for loop is assigned to thread 5
28 iteration of the for loop is assigned to thread 5
29 iteration of the for loop is assigned to thread 4
30 iteration of the for loop is assigned to thread 4
31 iteration of the for loop is assigned to thread 5
32 iteration of the for loop is assigned to thread 5
33 iteration of the for loop is assigned to thread 4
34 iteration of the for loop is assigned to thread 4
35 iteration of the for loop is assigned to thread 5
36 iteration of the for loop is assigned to thread 5
37 iteration of the for loop is assigned to thread 4
38 iteration of the for loop is assigned to thread 4
39 iteration of the for loop is assigned to thread 5
40 iteration of the for loop is assigned to thread 5
41 iteration of the for loop is assigned to thread 4
42 iteration of the for loop is assigned to thread 4
43 iteration of the for loop is assigned to thread 5
44 iteration of the for loop is assigned to thread 5
45 iteration of the for loop is assigned to thread 4
46 iteration of the for loop is assigned to thread 4
47 iteration of the for loop is assigned to thread 4
48 iteration of the for loop is assigned to thread 4
49 iteration of the for loop is assigned to thread 5

6. **Guided scheduling with chunk size not specified:**

Enter a, b, and n
1 100 50

With n = 50 trapezoids, our estimate
of the integral from 1.000000 to 100.000000 = 3.33397686600000e+05
1 iteration of the for loop is assigned to thread 4
2 iteration of the for loop is assigned to thread 4
3 iteration of the for loop is assigned to thread 4
4 iteration of the for loop is assigned to thread 4
5 iteration of the for loop is assigned to thread 5
6 iteration of the for loop is assigned to thread 5
7 iteration of the for loop is assigned to thread 5
8 iteration of the for loop is assigned to thread 4
9 iteration of the for loop is assigned to thread 4
10 iteration of the for loop is assigned to thread 4
11 iteration of the for loop is assigned to thread 4
12 iteration of the for loop is assigned to thread 4
13 iteration of the for loop is assigned to thread 4
14 iteration of the for loop is assigned to thread 5
15 iteration of the for loop is assigned to thread 5
16 iteration of the for loop is assigned to thread 5
17 iteration of the for loop is assigned to thread 4
18 iteration of the for loop is assigned to thread 4
19 iteration of the for loop is assigned to thread 4
20 iteration of the for loop is assigned to thread 5
21 iteration of the for loop is assigned to thread 5
22 iteration of the for loop is assigned to thread 4
23 iteration of the for loop is assigned to thread 4
24 iteration of the for loop is assigned to thread 5
25 iteration of the for loop is assigned to thread 5
26 iteration of the for loop is assigned to thread 5
27 iteration of the for loop is assigned to thread 5
28 iteration of the for loop is assigned to thread 5
29 iteration of the for loop is assigned to thread 5
30 iteration of the for loop is assigned to thread 4
31 iteration of the for loop is assigned to thread 4
32 iteration of the for loop is assigned to thread 5
33 iteration of the for loop is assigned to thread 5
34 iteration of the for loop is assigned to thread 4
35 iteration of the for loop is assigned to thread 4
36 iteration of the for loop is assigned to thread 5
37 iteration of the for loop is assigned to thread 4
38 iteration of the for loop is assigned to thread 4
39 iteration of the for loop is assigned to thread 4
40 iteration of the for loop is assigned to thread 4
41 iteration of the for loop is assigned to thread 4
42 iteration of the for loop is assigned to thread 5

43 iteration of the for loop is assigned to thread 4
44 iteration of the for loop is assigned to thread 5
45 iteration of the for loop is assigned to thread 4
46 iteration of the for loop is assigned to thread 5
47 iteration of the for loop is assigned to thread 4
48 iteration of the for loop is assigned to thread 5
49 iteration of the for loop is assigned to thread 5

7. **Guided scheduling with chunk size = 2:**

Enter a, b, and n
1 100 50
With n = 50 trapezoids, our estimate
of the integral from 1.000000 to 100.000000 = 3.33397686600000e+05
1 iteration of the for loop is assigned to thread 5
2 iteration of the for loop is assigned to thread 5
3 iteration of the for loop is assigned to thread 5
4 iteration of the for loop is assigned to thread 5
5 iteration of the for loop is assigned to thread 4
6 iteration of the for loop is assigned to thread 4
7 iteration of the for loop is assigned to thread 4
8 iteration of the for loop is assigned to thread 4
9 iteration of the for loop is assigned to thread 4
10 iteration of the for loop is assigned to thread 4
11 iteration of the for loop is assigned to thread 4
12 iteration of the for loop is assigned to thread 4
13 iteration of the for loop is assigned to thread 4
14 iteration of the for loop is assigned to thread 4
15 iteration of the for loop is assigned to thread 4
16 iteration of the for loop is assigned to thread 4
17 iteration of the for loop is assigned to thread 5
18 iteration of the for loop is assigned to thread 5
19 iteration of the for loop is assigned to thread 5
20 iteration of the for loop is assigned to thread 4
21 iteration of the for loop is assigned to thread 4
22 iteration of the for loop is assigned to thread 4
23 iteration of the for loop is assigned to thread 4
24 iteration of the for loop is assigned to thread 5
25 iteration of the for loop is assigned to thread 5
26 iteration of the for loop is assigned to thread 4
27 iteration of the for loop is assigned to thread 4
28 iteration of the for loop is assigned to thread 5
29 iteration of the for loop is assigned to thread 5
30 iteration of the for loop is assigned to thread 4
31 iteration of the for loop is assigned to thread 4
32 iteration of the for loop is assigned to thread 5

33 iteration of the for loop is assigned to thread 5
34 iteration of the for loop is assigned to thread 4
35 iteration of the for loop is assigned to thread 4
36 iteration of the for loop is assigned to thread 5
37 iteration of the for loop is assigned to thread 5
38 iteration of the for loop is assigned to thread 4
39 iteration of the for loop is assigned to thread 4
40 iteration of the for loop is assigned to thread 5
41 iteration of the for loop is assigned to thread 5
42 iteration of the for loop is assigned to thread 4
43 iteration of the for loop is assigned to thread 4
44 iteration of the for loop is assigned to thread 4
45 iteration of the for loop is assigned to thread 4
46 iteration of the for loop is assigned to thread 5
47 iteration of the for loop is assigned to thread 5
48 iteration of the for loop is assigned to thread 4
49 iteration of the for loop is assigned to thread 4

8. **Auto scheduling:**

Enter a, b, and n
1 100 50
With n = 50 trapezoids, our estimate
of the integral from 1.000000 to 100.000000 = 3.33397686600000e+05
1 iteration of the for loop is assigned to thread 0
2 iteration of the for loop is assigned to thread 0
3 iteration of the for loop is assigned to thread 0
4 iteration of the for loop is assigned to thread 0
5 iteration of the for loop is assigned to thread 1
6 iteration of the for loop is assigned to thread 1
7 iteration of the for loop is assigned to thread 1
8 iteration of the for loop is assigned to thread 2
9 iteration of the for loop is assigned to thread 2
10 iteration of the for loop is assigned to thread 2
11 iteration of the for loop is assigned to thread 3
12 iteration of the for loop is assigned to thread 3
13 iteration of the for loop is assigned to thread 3
14 iteration of the for loop is assigned to thread 4
15 iteration of the for loop is assigned to thread 4
16 iteration of the for loop is assigned to thread 4
17 iteration of the for loop is assigned to thread 5
18 iteration of the for loop is assigned to thread 5
19 iteration of the for loop is assigned to thread 5
20 iteration of the for loop is assigned to thread 6
21 iteration of the for loop is assigned to thread 6
22 iteration of the for loop is assigned to thread 6

23 iteration of the for loop is assigned to thread 7
24 iteration of the for loop is assigned to thread 7
25 iteration of the for loop is assigned to thread 7
26 iteration of the for loop is assigned to thread 8
27 iteration of the for loop is assigned to thread 8
28 iteration of the for loop is assigned to thread 8
29 iteration of the for loop is assigned to thread 9
30 iteration of the for loop is assigned to thread 9
31 iteration of the for loop is assigned to thread 9
32 iteration of the for loop is assigned to thread 10
33 iteration of the for loop is assigned to thread 10
34 iteration of the for loop is assigned to thread 10
35 iteration of the for loop is assigned to thread 11
36 iteration of the for loop is assigned to thread 11
37 iteration of the for loop is assigned to thread 11
38 iteration of the for loop is assigned to thread 12
39 iteration of the for loop is assigned to thread 12
40 iteration of the for loop is assigned to thread 12
41 iteration of the for loop is assigned to thread 13
42 iteration of the for loop is assigned to thread 13
43 iteration of the for loop is assigned to thread 13
44 iteration of the for loop is assigned to thread 14
45 iteration of the for loop is assigned to thread 14
46 iteration of the for loop is assigned to thread 14
47 iteration of the for loop is assigned to thread 15
48 iteration of the for loop is assigned to thread 15
49 iteration of the for loop is assigned to thread 15

The default assignment of iterations is approximately n/thread_count per thread when no scheduling method is specified.

In a guided schedule, each thread executes a chunk of iterations, and when a thread finishes a chunk, it requests another one. However, in a guided schedule, as chunks are completed, the size of the new chunks decreases.

Ex 5.10 page 265

Code:

```
/* File:    5.10.c
 * Prabhat Bhootra
 */

#include <stdio.h>
```

```
#include <stdlib.h>
#include <math.h>
#include <omp.h>


int main(int argc, char* argv[]) {
  int thread_count, n;
  double start, finish;
  thread_count = strtol(argv[1], NULL, 10);
  n = strtol(argv[2], NULL, 10);
  start = omp_get_wtime();
#  pragma omp parallel num_threads(thread_count)
  {
    int i;
    double my_sum = 0.0;
    for(i = 0; i < n; i++) {
#       pragma omp atomic
      my_sum += sin(i);
    }
  }
  finish = omp_get_wtime();
  printf("Thread_count = %d, n = %d, Time = %e seconds\n",
      thread_count, n, finish-start);
  return 0;
}
```

Test Results:

Thread Count	N	Time (seconds)
1	100	1.405890e-04
2	100	2.331740e-04
3	100	8.740662e-03

Based on these results, we can see that different executions of my sum += sin(i) are all treated as a single critical section. This implementation of OpenMP does not allow simultaneous execution of updates to different variables when the updates are protected by atomic directives.